Hochschule Wismar
University of Applied Sciences Technology, Business and Design
Fakultät für Ingenieurwissenschaften

Master-Thesis

Konzeption eines Vorgehensmodells zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme

Eingereicht am: 28. November 2022

von: Lukas Zorn

Betreuerin: Prof. Dr.-Ing. Antje Raab-Düsterhöft
Zweitbetreuer: Prof. Dr.-Ing. habil. Andreas Ahrens

Aufgabenstellung

Im Rahmen der Master-Thesis ist ein Vorgehensmodell zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme zu konzipieren. Zu diesem Zweck sind bestehende Vorgehensmodelle aus dem Bereich der relationalen Datenbanksysteme auf ihre Übertragbarkeit auf nicht-relationale Datenbanksysteme zu untersuchen. Ferner sind, soweit vorhanden, die CIS- und STIGs-Leitfäden beider Datenbanktypen in die Untersuchung einzubeziehen. Darüber hinaus können auch andere Best-Practice-Ansätze sowie weitere Leitfäden, die sich im Laufe der Bearbeitung als relevant erweisen, berücksichtigt werden. Anschließend sind aus dieser Analyse die Ziele eines Härtungsprozesses abzuleiten, z. B. die Sicherstellung von Authentifizierung und Autorisierung, sowie Methoden zu deren Umsetzung aufzuzeigen, z. B. die Aktivierung eines entsprechenden Moduls/Plug-ins des Datenbanksystems. Das daraus entstehende Vorgehensmodell stellt das erste Zwischenergebnis der Master-Thesis dar.

Das Vorgehensmodell ist auf dieser Grundlage anhand von mindestens zwei nicht-relationalen Datenbanksystemen, deren Auswahlprozess im Rahmen der Master-Thesis darzulegen ist, praktisch anzuwenden. Die Entwicklung der Härtungsmaßnahmen ist anhand der zuvor entwickelten Methoden mit dem Ziel durchzuführen, konkrete technische Maßnahmen für das jeweilige Datenbanksystem zu formulieren. Die konkreten technischen Härtungsmaßnahmen stellen das zweite Zwischenergebnis der Master-Thesis dar.

Abschließend ist für mindestens eins der ausgewählten Datenbanksysteme ein Progress Chef InSpec Compliance-Profil zu entwickeln. Das Ziel der Entwicklung des Compliance-Profils ist die automatisierte Überprüfung eines bestehenden Datenbanksystems hinsichtlich seiner Konformität zu den entwickelten Härtungsmaßnahmen. Optional wird das Compliance-Profil nach Abschluss der Master-Thesis als Open Source veröffentlicht, z. B. auf der Plattform GitHub.

Kurzfassung

Im Rahmen dieser Master-Thesis wird ein Vorgehensmodell zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme auf der Basis mehrerer etablierter Standards verschiedener Organisationen entwickelt. Diese bestehen zum einen aus den übergreifenden prozess- und systemorientierten BSI IT-Grundschutz-Bausteinen und zum anderen aus 3 CIS-Benchmarks und 2 STIGs-Leitfäden, die sich bei beiden explizit mit der technischen Absicherung eines bestimmten nicht-relationalen Datenbanksystems befassen. Deren individuelle Maßnahmen werden zu insgesamt 82 neuen Anforderungen zusammengefasst, für die jeweils eine detaillierte Beschreibung formuliert und, wo möglich, weitere Best-Practice-Ansätze als Leitfaden für die technische Realisierung angegeben werden.

Anschließend wird das Vorgehensmodell auf die Community-Editionen der Datenbanksysteme Neo4j und Redis angewendet, wodurch die jeweiligen technischen Maßnahmen zur Einrichtung einer gehärteten Konfiguration im Detail herausgearbeitet werden. Für Redis werden diese zusätzlich in ein InSpec-Compliance-Profil überführt, dessen Ausführung einen automatisierten Soll-Ist-Abgleich der Konfiguration im Hinblick auf die Konformität mit den Anforderungen ermöglicht.

Grundsätzlich beschränkt sich der Geltungsbereich der Master-Thesis auf technische Maßnahmen, während organisatorische und prozessuale Aspekte nicht berücksichtigt werden.

Abstract

In the course of this master thesis, a process model for the derivation of hardening procedures for non-relational database systems is developed on the basis of several established standards of different organizations. These consist, on the one hand, of the generic process- and system-oriented BSI IT-Grundschutz blocks and, on the other hand, of 3 CIS Benchmarks and 2 STIGs Guides, both of which explicitly deal with the technical hardening of a specific non-relational database system. Their individual measures are combined into a total of 82 new requirements, for each of which a detailed description is expressed and, where possible, further best-practice approaches are given as a guide to technical implementation.

The process model is then applied to the community editions of the Neo4j and Redis database systems, whereby the respective technical measures for setting up a hardened configuration are worked out in detail. For Redis, these are additionally converted into an InSpec compliance baseline, whose execution enables an automated target-performance comparison of the configuration with regard to its conformity with the requirements.

As a rule, the scope of the master thesis is limited to technical measures, while organizational and process-related aspects are not taken into account.

Inhaltsverzeichnis

1 Einleitung

Zunehmend ausgefeiltere Cyber-Bedrohungen haben sich in den letzten Jahren zu einem schwerwiegenden Gegner in der immer größer werdenden Welt des Cyber-Raums herausgebildet. Laut einer vom Digitalverband Bitkom in Auftrag gegebenen Studie vom 31. August 2022 entsteht der deutschen Wirtschaft durch Diebstahl von IT-Geräten und Daten, Spionage und Sabotage ein jährlicher Schaden von rund 203 Milliarden Euro [1]. Im Zeitraum 2018/2019 betrug der finanzielle Schaden 103 Milliarden Euro, was somit einer Zunahme von $\approx 97,09$ % innerhalb von etwa drei Jahren entspricht [2]. Zu den häufigsten Angriffsarten zählen dabei Infektionen mit Schadprogrammen (25 %), Structured Query Language (SQL)-Injektionen (14 %) und Ransomware-Angriffe (12 %), wie eine Umfrage bei Unternehmen in Deutschland vom 10. Januar bis 13. März 2022 ergab [3].

Auch das Bundeskriminalamt (BKA) verzeichnet in seinen Bundeslagebildern zur Cyberkriminalität eine zunehmende Verlagerung in den digitalen Raum. Im Berichtszeitraum 2020 stiegen die Straftaten der Kategorie „Cybercrime im weiteren Sinne“ um $8,7$ % gegenüber dem Vorjahr 2019 [4]. Darunter fallen Straftaten, die unter Zuhilfenahme von Informationstechnologien verübt werden, aber auch unabhängig davon erfolgen können, wie z. B. Betrugsdelikte [5]. Im Gegensatz dazu stieg die Anzahl der Straftaten der Kategorie „Cybercrime im engeren Sinne“ um $7,9$ %. Auch im Berichtszeitraum 2021 nahmen die Straftaten nochmals erheblich zu. Vor diesem Hintergrund und der daraus weiter gewachsenen Bedeutung wurden die bisherigen Cybercrime-Kategorien zusammengefasst und ein gesamtheitlicher Anstieg von $12,2$ % gegenüber dem Vorjahr ausgewiesen [6]. Gleichzeitig befindet sich die Aufklärungsquote unverändert auf einem niedrigen Niveau und sank nochmals um $2,7$ % auf einen Wert von $29,3$ %. Eine Trendwende ist daher zum jetzigen Zeitpunkt nicht zu erwarten.

Gleichzeitig wird die Vernetzung von Infrastrukturen immer stärker vorangetrieben, was nicht zuletzt auch durch die Corona-Pandemie und dem damit einhergehenden Nachfragezuwachs an Home-Office-Kapazitäten stark begünstigt wurde [7]. Moderne IT-Infrastrukturen stellen verteilte Systeme mit einer zunehmenden vertikalen

und horizontalen Vernetzung dar, die durch den Einsatz von cloudbasierten Lösungen keine statischen Sicherheitsgrenzen mehr aufweisen [8]. Dies führt nicht nur zu einem Anstieg der Gesamtkomplexität, sondern auch zu einer erheblichen Vergrößerung der potenziellen Angriffsfläche. Im Zentrum dieser Entwicklung steht eine ebenfalls immer größer werdende Bandbreite an Anwendungen. Deren Fundament stellt fast immer eine Datenbank zur Speicherung, Verarbeitung und Bereitstellung von Anwendungsdaten dar. Das Spektrum dieser ist vielfältig und reicht von sensiblen Mitarbeiterdaten bis hin zu kritischen internen Geschäftsdaten. Die Absicherung dieser Daten ist daher von entscheidender Bedeutung, zumal die Bedrohung durch Cyber-Erpressungen seit dem russischen Angriffskrieg auf die Ukraine nochmals deutlich zugenommen hat [9].

Konventionelle relationale Datenbanksysteme werden mit Blick auf die Entwicklung der Datenbanksprache SQL bereits seit den 1970er Jahren eingesetzt. Im Gegensatz dazu ist der Begriff NoSQL erst zu Beginn des 21. Jahrhunderts geprägt worden [10]. Entsprechende nicht-relationale Datenbanken (NoSQL) werden jedoch zunehmend in Big-Data- sowie Echtzeit-Webanwendungen eingesetzt und haben in den letzten Jahren aufgrund rasant steigender Datenmengen immer mehr an Bedeutung gewonnen [11]. Im Gegensatz zu relationalen Datenbanksystemen, die somit seit mehreren Jahrzehnten weit verbreitet sind, gibt es für NoSQL-Datenbanksysteme bislang kaum Härtungsansätze, die über die allgemeinen Empfehlungen zum produktiven Einsatz des Herstellers/Entwicklers hinausgehen. Insbesondere vor dem Hintergrund der drei eingangs erwähnten Angriffsarten, die in direktem Bezug zur Datenbanksicherheit stehen, und der rasanten Zunahme der Cyberkriminalität bei gleichzeitig sinkender Aufklärungsquote muss die Absicherung nicht-relationaler Datenbanken in den Vordergrund rücken. Zu diesem Zweck kann eine Härtungsstrategie entworfen und umgesetzt werden. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) beschreibt die Härtung als „[...] die Entfernung aller Softwarebestandteile und Funktionen, die zur Erfüllung der vorgesehenen Aufgabe durch das Programm nicht zwingend notwendig sind“ [12]. Obgleich diese Definition allgemein gültig ist, reicht sie nicht aus, um eine umfassende Härtungsstrategie für nicht-relationale Datenbanken zu entwickeln. Vielmehr muss die vom BSI definierte Methodik weitreichend ergänzt werden.

Die Konzeption eines Vorgehensmodells zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme ist daher von elementarer Bedeutung für die Gewährleistung der Informationssicherheit moderner Anwendungen und wird nachfolgend in dieser Master-Thesis thematisiert. Dazu werden im folgenden Kapitel 2

zunächst die dafür notwendigen technischen Grundlagen behandelt, bevor die Entwicklung des Vorgehensmodells in Kapitel 3 umgesetzt und anschließend in Kapitel 4 beispielhaft anhand von zwei Datenbanksystemen angewendet sowie in ein InSpec Compliance-Profil überführt wird. In Kapitel 5 folgt eine Auswertung im Hinblick auf den Erfüllungsgrad der beiden Datenbanksysteme sowie eine Auseinandersetzung mit weiteren Evaluierungsfragen. Zuletzt wird im Kapitel 6 das Fazit in Form einer Zusammenfassung und Einordnung präsentiert.

2 Technische Grundlagen

Das Ziel dieses Kapitels ist die Vermittlung der technischen Grundlagen, die bei der Entwicklung und späteren Anwendung des Vorgehensmodells im Vordergrund stehen. Dazu werden zunächst die Grundprinzipien von relationalen und nicht-relationalen Datenbankmanagementsystemen kurz erläutert. Im Anschluss daran werden die für die Entwicklung des Vorgehensmodells herangezogenen Standards vorgestellt. Abschließend folgt ein Überblick über die Entwicklung von Compliance-Profilen mit Progress Chef InSpec, mit dem eine automatisierte Compliance-Prüfung der im Rahmen der Anwendung identifizierten Maßnahmen implementiert wird.

2.1 Datenbanken und Datenbankmanagementsysteme

Die Begriffe Datenbank und Datenbankmanagementsystem (DBMS) werden oftmals synonym verwendet, obwohl es sich um zwei gänzlich abweichende Bedeutungen handelt, die klar unterschieden werden sollten.

Eine Datenbank beschreibt ein elektronisches System, das zur Speicherung, Verarbeitung und Bereitstellung von Informationen eingesetzt wird. Dabei handelt es sich im Allgemeinen um eine klar definierte Zusammenstellung bzw. Verknüpfung von Datensätzen oder Dateien unterschiedlichen Typs, die auf diese Weise eine organisierte Sammlung von Informationen darstellen [13]. Üblicherweise werden diese auf einem IT-System oder einer anderen Art von Hardware gespeichert.

Ein DBMS hingegen ist eine Anwendung, die den Zugriff auf, die Interaktion mit und die Bearbeitung von Datenbankinhalten ermöglicht. Diese fungiert folglich als anwendungsunabhängige Schnittstelle zur Datenbank und stellt in diesem Zusammenhang dem Benutzer oder der zugehörigen Anwendung bzw. Programmierschnittstelle (API) einige zentrale Funktionen über eine Speicher- und eine Verwaltungskomponente zur Verfügung [14]. Unabhängig vom verwendeten Datenbankmodell umfassen diese Funktionen in erster Linie [15]:

- Die organisierte Speicherung der Daten und ihrer Metadaten,
- Der Abruf, die Aktualisierung oder die Löschung der Daten mittels einer Abfrage- und Datenmanipulationssprache,
- Die Durchsetzung von Beschränkungen, anhand derer sichergestellt wird, dass die Daten bestimmten Regeln entsprechen, wie z. B. eine Bereichsintegrität,
- Die Sicherstellung von Authentisierung, Authentifizierung und Autorisierung,
- Die Unterstützung von Transaktionen und Nebenläufigkeit bzw. Mehrbenutzerfähigkeit zur Datensicherheit,
- Die Implementierung von Mechanismen zur Detektion und Beseitigung von Datenbankfehlern.

Unter Einbeziehung des verwendeten Datenbankmodells können einige der aufgeführten Funktionen jedoch auch entfallen oder nicht umfassend implementiert sein. Dies ist insbesondere dann der Fall, wenn sie dem primären Anwendungszweck des Datenbankmodells zuwiderlaufen, wie beispielsweise der Realisierung einer extrem leistungsstarken Datenverarbeitung (Performance) [16]. Auf Basis der unterstützten Datenbankmodelle werden DBMS daher unterschieden. Eine gängige Klassifizierung stellt die Einteilung in relationale und nicht-relationale Datenbanksysteme dar.

2.1.1 Relationale Datenbanksysteme

Das relationale Modell ist Stand Januar 2022 nach wie vor das beliebteste Datenbankmodell, gemessen am absoluten Ranking nach der Kategorie des entsprechenden DBMS [17].

Das Grundprinzip des relationalen Modells ist das Informationsprinzip, demzufolge alle Informationen durch Datenwerte in Beziehungen (Relationen) dargestellt werden. Dazu werden Informationen in mehreren Tabellen gespeichert, die über verschiedene Arten von Relationen miteinander in Verbindung gesetzt werden. Gemäß diesem Prinzip ist eine relationale Datenbank eine Menge von Relationen, und das Ergebnis jeder Abfrage wird ebenfalls als eine Relation dargestellt [18]. Zur Bearbeitung von Daten in relationalen Datenbanken oder zur Durchführung von Abfragen wird die Datenbanksprache SQL verwendet. Ihre Grundlage ist die relationale Algebra, anhand derer sich die Tabellen aus den folgenden Grundelementen zusammensetzen [19]:

Tupel Eine Tabelle besteht aus mehreren Datensätzen, die jeweils durch ein Tupel (Tabellenzeile) repräsentiert werden.

Attribut Jedes Tupel setzt sich wiederum aus einer Reihe von Attributen (Tabellenspalte) zusammen. Jedes Attribut muss eine eindeutige Bezeichnung innerhalb der Tabelle und einen vordefinierten Datentyp, z. B. „String“, „Integer“ oder auch „Datum und Uhrzeit“, aufweisen.

Relationsschema Das Relationsschema hält die Anzahl und den Typ aller Attribute für jede Tabelle in einer Datenbank fest. Jedes Tupel entspricht demnach einer individuellen Instanz dieses Schemas.

Constraints Mit Hilfe von Constraints (Beschränkungen) werden bei der Definition des Relationsschemas Integritätsregeln festgelegt, die den Wertebereich einschränken oder die Qualität und dadurch die Zuverlässigkeit der Daten sicherstellen [20]. Hervorzuheben sind die Beschränkungen NOT NULL, das sicherstellt, dass ein Attribut nie keinen Wert (NULL) annehmen kann [21], und UNIQUE, das sicherstellt, dass sämtliche Werte eines Attributs für jedes Tupel einer Tabelle eindeutig sind [22]. Aus der Kombination der Eigenschaften dieser beiden Beschränkungen ergibt sich der Constraint PRIMARY KEY (Primärschlüssel), der für die eindeutige Zuordnung von Informationen aus mehreren Tabellen verwendet wird. Die Referenzierung des Constraints PRIMARY KEY in einer anderen Tabelle wird als FOREIGN KEY (Fremdschlüssel) bezeichnet und stellt auf diese Weise eine Verbindung zwischen beiden Tabellen her [23]. Dadurch wird verhindert, dass ungültige Daten in das Fremdschlüsselattribut der referenzierten Tabelle eingefügt werden, indem die Eingabe gültiger Werte erzwungen wird.

Zur Veranschaulichung der Grundelemente wurde mit der Programmiersprache Python auf die öffentliche API von „Have I Been Pwned?“ [24] zugegriffen, um alle kompromittierten Datensätze mit Stand vom 8. August 2022 in zwei zusammenhängende Relationsschemata zu gruppieren. Das zugehörige Python-Skript, welches im Folgenden auch für die Generierung von Beispielen für nicht-relationale Datenbankmodelle verwendet wird, kann dem Anhang A entnommen werden.

GID	Name des Unternehmens	Domain
1	Amart Furniture	`amartfurniture.com.au`
2	BlackBerry Fans	`blackberryfans.org`
3	CDEK	`cdek.ru`
4	Doxbin	`doxbin.com`
5	Fanpass	`fanpass.co.uk`

Tabelle 1: Gekürzte aufbereitete Ausgabe eines Relationsschemas „Geschädigter“ aus der „Have I Been Pwned?“ API als Beispiel für relationale Datenbanken

Die Tabelle 1 beschreibt durch das gekürzt dargestellte Relationsschema den Datensatz der Geschädigten unter der Angabe einer eindeutigen ID (PRIMARY KEY), dem Namen des betroffenen Unternehmens und der zugehörigen Domain. Die vollständige Ausgabe kann in Anhang B eingesehen werden.

DID	GID	Datum der Sicherheitsverletzung	Anzahl betroffener Konten	Art der kompromittierten Daten
1	4	2022-01-05	370.794	Browser user agent details, Email addresses, Passwords, Usernames
2	3	2022-03-09	19.218.203	Email addresses, Names, Phone numbers
3	5	2022-04-30	112.251	Email addresses, Genders, Names, Partial dates of birth, Passwords, Phone numbers, Physical addresses, Purchases, Social media profiles
4	2	2022-05-06	174.168	Email addresses, IP addresses, Passwords, Usernames
5	1	2022-05-16	108.940	Email addresses, Names, Passwords, Phone numbers, Physical addresses

Tabelle 2: Gekürzte aufbereitete Ausgabe eines Relationsschemas „Kompromittierte Datensätze“ aus der „Have I Been Pwned?“ API als Beispiel für relationale Datenbanken

Die Tabelle 2 wiederum beschreibt, durch das dargestellte Relationsschema, den Datensatz der abgeflossenen „kompromittierten“ Datensätze. Zu diesem Zweck werden das Datum der Sicherheitsverletzung, die zum entsprechenden Datensatz (Tupel) zugehörige ID des Geschädigten (FOREIGN KEY), die Anzahl betroffener Konten sowie die Art der kompromittierten Daten angegeben. Die vollständige Ausgabe kann ebenfalls in Anhang B eingesehen werden.

Durch die Referenzierung des kompromittierten Datensatzes mit der ID des Geschädigten werden beide Tabellen durch eine Eins-zu-Viele-Beziehung (1:N) verbunden. Die Art der Referenzierung ermöglicht eine redundanzfreie und durchgehend konsistente Speicherung, sodass im Ergebnis eine Normalform der Gesamtdaten entsteht [18]. Die Tabelle 2 mit dem Fremdschlüssel wird als untergeordnete Tabelle und die Tabelle 1 mit dem Primärschlüssel als referenzierte oder übergeordnete Tabelle bezeichnet [25].

Zusammenfassend kann festgestellt werden, dass der Vorteil von relationalen Datenbanksystemen in der Normalisierung der Gesamtdaten liegt. Die daraus abgeleiteten Ziele in Form der Inkonsistenz- und Redundanzvermeidung und der daraus resultierenden eindeutigen und vordefinierten Strukturierung der Daten sorgen jedoch auch für einen zunehmenden Mangel an Flexibilität [26]. Verstärkt wird dieser Umstand durch die im Laufe der Zeit ständig wachsenden Datenbestände in Kombination mit immer komplexer werdenden Relationen. Deren nachträgliche Entflechtung im Falle einer notwendigen Neustrukturierung des Datenbestands ist zumeist sehr aufwendig und zeitintensiv. Der Einsatz von relationalen DBMS ist daher insbesondere im Bereich von Big-Data für die Verarbeitung großer Datenmengen ungeeignet, da diese zumeist keine vordefinierte Struktur aufweisen und somit auf eine größtmögliche Flexibilität durch das DBMS angewiesen sind [27]. Auch der Umgang mit unstrukturierten Datensätzen wie Dokumenten, Dateien oder Bildern ist in vielen Anwendungsbereichen unerlässlich. Zusätzlich sind relationale Datenbanken bei der Verarbeitung sehr großer Datenmengen im direkten Vergleich nicht sehr leistungsfähig. Nicht-relationale DBMS (NoSQL-Datenbanken) sind für die Verarbeitung dieser Art von Daten hingegen besonders geeignet und werden im folgenden Abschnitt vorgestellt.

2.1.2 Nicht-relationale Datenbanksysteme

Die Abkürzung NoSQL steht für „not only SQL“ oder, inzwischen seltener, „non-SQL“ oder auch „non-relational“. Letztere beschreiben definitionsgemäß einen alternativen

Ansatz zur Speicherung, Verarbeitung und Bereitstellung von Daten, der nicht auf der Modellierung von tabellarischen Relationen basiert. Heute steht NoSQL hingegen für eine viel umfangreichere Anzahl an Datenbankmodellen. Aufgrund der Ausschlussdefinition und der am Verwendungszweck orientierten Entwicklung existiert jedoch kein einheitlicher Gegenentwurf, sondern umfasst ausdrücklich alle Datenbankmodelle, die nicht oder, im Falle hybrider Lösungen, nicht nur auf SQL basieren [28]. Zum heutigen Zeitpunkt existieren daher mehr als 300 Datenbanksysteme, die diese Definition erfüllen [29]. Dennoch können diese in die folgenden vier Hauptkategorien untergliedert werden:

Dokumentenorientierte Datenbank Eine dokumentenorientierte Datenbank (englisch „Document-oriented database“) ordnet Daten in schemalosen Zusammenstellungen an, die als Dokumente bezeichnet werden. Der Aufbau der Dokumente muss im Gegensatz zu relationalen Datenbanken somit keine vordefinierten Struktur aufweisen. Aufgrund dieser weitgehenden Schemafreiheit bestehen keine Relationen zwischen den Dokumenten untereinander [30]. Die inhaltliche Vernetzung von Einzelinformationen erfolgt durch den Entwickler auf der Anwendungsebene, wird also nicht durch das Datenbankmodell selbst realisiert [31].

Die fünf populärsten dokumentenorientierten Datenbanken sind Stand November 2022 – in absteigender Reihenfolge – MongoDB, Amazon DynamoDB, Databricks, Microsoft Azure Cosmos DB und Couchbase [32].

Das Listing 1 zeigt als Beispiel für dokumentenorientierte Datenbanken eine gekürzte aufbereitete Ausgabe aus der „Have I Been Pwned?“ API, wie zuvor bereits die Tabellen 1 und 2. Im Gegensatz zum tabellarischen Ansatz wurden alle Relationen aufgelöst und jeder kompromittierte Datensatz in ein separates Dokument übertragen. Die vollständige Ausgabe kann in Anhang C eingesehen werden.

```
1  [
2      {
3          "Name des Unternehmens": "Twitter",
4          "Domain": "twitter.com",
5          "Datum der Sicherheitsverletzung": "2022-01-01",
6          "Anzahl betroffener Konten": 6682453,
7          "Art der kompromittierten Daten": [
8              "Bios",
9              "Email addresses",
10             "Geographic locations",
11             "Names",
12             "Phone numbers",
```

```
13              "Profile photos",
14              "Usernames"
15          ]
16      },
17      {...}
18  ]
```

Listing 1: Gekürzte aufbereitete Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für dokumentenorientierte Datenbanken

Schlüssel-Werte-Datenbank Eine Schlüssel-Werte-Datenbank (englisch „Key-value database“) speichert Daten in Zuordnungstabellen, welche mit assoziativen Datenfeldern (englisch „Dictionary“) verglichen werden können. Folglich wird jede zu speichernde Information (Wert) einem eindeutigen Schlüssel zugeordnet [30]. Im Vergleich zu relationalen Datenbanken müssen diese jedoch nicht einen bestimmten Datentyp oder eine bestimmte Länge einhalten. Aus Performance-Gründen ist es jedoch nicht ratsam, diese Flexibilität bei der Vergabe von Schlüsseln vollständig in Anspruch zu nehmen, da dies den Abruf von Informationen unnötig verlangsamt. Werden sehr lange Schlüssel benötigt, sollte ein Hash-Algorithmus verwendet werden, um die Länge zu reduzieren [33].

Die fünf populärsten Schlüssel-Werte-Datenbanken sind Stand November 2022 – in absteigender Reihenfolge – Redis, Amazon DynamoDB, Microsoft Azure Cosmos DB, Memcached und Hazelcast [34].

Das Listing 2 zeigt als Beispiel für Schlüssel-Werte-Datenbanken ebenfalls eine gekürzte aufbereitete Ausgabe aus der „Have I Been Pwned?“ API. Dazu wurden die Dokumente aus dem obigen Beispiel für dokumentenorientierte Datenbanken in Redis-Prozeduren überführt. Die vollständige Ausgabe kann in Anhang D eingesehen werden.

```
1 SET name-des-unternehmens#1 "Twitter"
2 SET domain#1 "twitter.com"
3 SET datum-der-sicherheitsverletzung#1 "2022-01-01"
4 SET anzahl-betroffener-konten#1 "6682453"
5 SADD art-der-kompromittierten-daten#1 "Bios"
6 SADD art-der-kompromittierten-daten#1 "Email addresses"
7 SADD art-der-kompromittierten-daten#1 "Geographic locations"
8 SADD art-der-kompromittierten-daten#1 "Names"
9 SADD art-der-kompromittierten-daten#1 "Phone numbers"
```

```
10 SADD art-der-kompromittierten-daten#1 "Profile photos"
11 SADD art-der-kompromittierten-daten#1 "Usernames"
```

Listing 2: Gekürzte aufbereitete Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für Schlüssel-Werte-Datenbanken

Graphdatenbank Eine Graphdatenbank (englisch „Graph database“) werden Daten in Knoten und Kanten organisiert, denen jeweils Eigenschaften zugeordnet werden können [35]. Jeder Knoten stellt eine Entität in der Graphdatenbank dar und kann mit einer Typenbezeichnung versehen werden, z. B. „Datensatz“ oder „Unternehmen“. Neben der Bezeichnung können jedem Knoten beliebig viele Eigenschaften in Form von Schlüssel-Wert-Paaren zugewiesen werden (vgl. Schlüssel-Werte-Datenbanken) [30]. Knoten können über Kanten in eine unidirektionale oder bidirektionale Beziehung zueinander gesetzt werden, z. B. „Datensatz“ ← ist zugehörig → „Unternehmen“ (Bidirektionalität). Wie Knoten können alle Kanten eine Bezeichnung und Eigenschaften in Form von Schlüssel-Wert-Paaren enthalten. Die Anzahl der Kanten, die mit einem Knoten verbunden werden können, ist unbegrenzt. Dies ermöglicht ein schnelles und effizientes Abrufen von Daten, da eine flexible Verknüpfung von Daten über den gesamten Datensatz hinweg ohne Leistungseinbußen erzielt werden kann [36]. Die Verwendung von Graphdatenbanken eignet sich daher besonders gut für stark intervernetzte Datenstrukturen.

Die fünf populärsten Graphdatenbanken sind Stand November 2022 – in absteigender Reihenfolge – Neo4j, Microsoft Azure Cosmos DB, Virtuoso, ArangoDB und OrientDB [37].

Spaltenorientierte Datenbank Eine spaltenorientierte Datenbank (englisch „Column oriented database“) kann am ehesten mit dem tabellarischen Ansatz relationaler Datenbanken verglichen werden. Der Unterschied besteht jedoch in der Speicherung aller Informationen in Spalten anstelle von Zeilen. Die Referenzierung der Daten erfolgt indirekt über die Nummer der Zeile. Ein vollständiger Einzeldatensatz kann somit durch Abruf der Informationen aus allen vorliegenden Spalten mit der gleichen Zeilennummer realisiert werden [30]. Die Organisation der Daten in Spalten ermöglicht einen wesentlich effizienteren Zugriff auf Teilinformationen im Form einzelner Attribute eines Gesamtdatenbestandes, da auf nicht relevante Spalten nicht zugegriffen werden muss. Die Zeitersparnis nimmt insbesondere bei extrem großen Datenmengen stark zu [38]. Darüber hinaus kann eine Datenkompression wesentlich effektiver umgesetzt werden, da moderne Kompressionsalgorithmen wie

Lempel-Ziv-Oberhumer (LZO), Lempel-Ziv-Vier (LZ4) oder Snappy besonders effektiv bei gleichförmigen Daten wirken, was bei einer spaltenbasierten Datenhaltung überwiegend der Fall ist. Diese Vorteile werden jedoch durch eine geringere Leistungsfähigkeit beim Hinzufügen neuer Daten konterkariert, da dazu der Zugriff auf alle Spalten erfolgen muss [39].

Die fünf populärsten spaltenorientierte Datenbanken sind Stand November 2022 – in absteigender Reihenfolge – Apache Cassandra, Apache HBase, Microsoft Azure Cosmos DB, Datastax Enterprise und Microsoft Azure Table Storage [40].

Die Tabelle 3 zeigt als Beispiel für spaltenorientierte Datenbanken erneut eine gekürzte aufbereitete Ausgabe aus der „Have I Been Pwned?“ API. Die vollständige Ausgabe kann in Anhang E eingesehen werden.

Name des Unternehmens	**Domain**	**Datum der Sicherheitsverletzung**	**...**
Twitter	`twitter.com`	2022-01-01	...
Doxbin	`doxbin.com`	2022-01-05	...
MacGeneration	`macg.co`	2022-01-29	...
GiveSendGo	`givesendgo.com`	2022-02-07	...
NVIDIA	`nvidia.com`	2022-02-23	...

Tabelle 3: Gekürzte aufbereitete Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für spaltenorientierte Datenbanken

Neben diesen vier Hauptkategorien finden sich noch viele andere NoSQL Datenbankmodelle, die einen gänzlich anderen oder hybriden Ansatz verfolgen, wie zum Beispiel die Folgenden [29]:

- Grid- und Cloud-Datenbank-Lösungen
- Mehrdimensionale Datenbank-Management-Systeme
- Multimodell-Datenbank-Management-Systeme
- Multivalente Datenbank-Management-Systeme
- Objekt-Datenbank-Management-Systeme
- XML-Datenbankverwaltungssysteme

Für jede der zuvor genannten Datenbankkategorien gibt es viele verschiedene Datenbanklösungen. Diese Vielfalt an Anwendungen, vor allem aber bezogen auf die Unterkategorien, verdeutlicht den problemorientierten Entwicklungsansatz von NoSQL-Datenbanken. Ein großer Teil der Datenbanksysteme wurde entwickelt, um ein ganz bestimmtes individuelles Problem zu lösen, das von den bereits vorhandenen Datenbanksystemen entweder nicht oder nicht ausreichend gelöst wurde. Der Nutzen für Anwendungsentwickler in diesem Zusammenhang steht hingegen der Entwicklung einer allgemeinen Vorgehensweise zur Ableitung von Konfigurations- und Compliance-Richtlinien entgegen. Während relationale Datenbanken im Kern über ein einheitliches Datenhaltungs- und Normalisierungskonzept einschließlich der domänenspezifischen Programmiersprache SQL verfügen, gibt es aufgrund des problem- und damit lösungsorientierten Entwicklungsansatzes kein einheitliches Konzept für NoSQL-Datenbanken, sondern allenfalls punktuelle Überschneidungen. Allerdings ist dies weder überraschend noch erstrebenswert, da beide Interessen im Widerspruch zueinander stehen.

Ein Teil dieser Master-Thesis wird es daher sein, Lösungsansätze und Handlungsempfehlungen für diesen Konflikt aufzuzeigen.

2.2 Anatomie von Konfigurations- und Compliance-Richtlinien

Um einen sicheren Betrieb zu gewährleisten, ist es notwendig, für alle Geschäftsprozesse, IT-Systeme und Informationen ein den Anforderungen entsprechendes Sicherheitsniveau zu schaffen. Ein holistisches Gesamtkonzept bildet die Grundlage und den Ausgangspunkt für den Aufbau eines belastbaren Information Security Management System (ISMS) [41]. Dabei handelt es sich um einen kontinuierlichen Prozess, bei dem Strategien und Maßnahmen ständig überprüft und an sich ändernde Anforderungen angepasst werden. Neben organisatorischen, personellen und konzeptionellen Aspekten muss im Rahmen eines funktionalen Gesamtkonzepts auch die Resilienz der verschiedenen IT-, Netz- und Infrastrukturkomponenten nachhaltig erhöht werden. Zu diesem Zweck werden Konfigurations- und Compliance- bzw. Konformitätsrichtlinien entwickelt, um im Vergleich zur Standardkonfiguration der Anwendung ein höheres Schutzniveau zu garantieren [42].

In diesem Kontext wird in diesem Abschnitt nachfolgend zunächst auf den BSI IT-Grundschutz eingegangen, auf dessen Grundlage die Einführung eines ISMS durchgeführt werden kann. Anschließend folgt eine kurze Einführung in die Technische

Richtlinien des BSI (BSI-TR) sowie die Center for Internet Security (CIS) Benachmarks und Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIGs), auf deren Grundlage eine Vielzahl von Konfigurations- und Compliance-Richtlinien veröffentlicht werden.

2.2.1 BSI IT-Grundschutz

Die erste Version des IT-Grundschutzhandbuchs wurde vom BSI in Zusammenarbeit mit führenden Wirtschaftsunternehmen 1994 veröffentlicht, drei Jahre nach seiner Gründung durch Inkrafttreten des BSI-Errichtungsgesetzes am 1. Januar 1991 [43]. Dabei handelte es sich um einen modularen Ansatz mit Bausteinen, die in einem umfangreichen Katalog zusammengestellt wurden, und auf deren Grundlage ab dem Jahr 2004 bereits eine Zertifizierung durch das BSI erfolgen konnte [44]. Die Absicherung nach BSI IT-Grundschutz wurde im Laufe der Zeit kontinuierlich weiterentwickelt und ausgebaut. Damit wurde ein Instrument geschaffen, um einen ganzheitlichen Ansatz zur Gewährleistung der Informationssicherheit auf technischer, infrastruktureller, organisatorischer und personeller Ebene abzuleiten [41]. Im Jahr 2006 wurde der Aufbau und die Struktur des IT-Grundschutzhandbuchs aufgebrochen und so an die für eine Zertifizierung nach ISO/IEC 27001 erforderlichen Anforderungen angepasst [43]. Zu diesem Zweck wurde die Methodik in einzelne BSI-Standards sowie die Bausteine zu den einzelnen Gefährdungen und Maßnahmen in die IT-Grundschutz-Kataloge (später ab dem Jahr 2018 IT-Grundschutz-Kompendium) aufgeteilt. Dadurch rückte die Umsetzung eines ISMS stärker in den Fokus. Seitdem ist eine Zertifizierung nach ISO/IEC 27001 für die Implementierung eines ISMS innerhalb einer Institution auch auf Basis von IT-Grundschutz möglich [45]. Die wichtigsten Veröffentlichungen und Standards, die aus dieser Entstehungsgeschichte hervorgegangen und heute gültig sind, sind die Folgenden [46]:

- BSI IT-Grundschutz-Kompendium: Werkzeug für Informationssicherheit
- BSI-Standard 200-1: Managementsysteme für Informationssicherheit (ISMS)
- BSI-Standard 200-2: IT-Grundschutz-Methodik
- BSI-Standard 200-3: Risikomanagement
- BSI-Standard 100-4: Notfallmanagement

Darüber hinaus wird derzeit der BSI-Standard 200-4 entwickelt, der voraussichtlich im Laufe des Jahres 2023 den bisherigen BSI-Standard 100-4 sukzessive ablösen wird

[47]. Damit wird der derzeitige Fokus auf das Notfallmanagement um das Thema eines ganzheitlichen Business Continuity Management (BCM) stark erweitert und modernisiert [48].

BSI Technische Richtlinien

Die BSI-Standards stellen bewährte Verfahren und das IT-Grundschutz-Kompendium konkrete Anforderungen bereit, ohne jedoch auf spezifische technische Mindestanforderungen an IT-Sicherheitsstandards einzugehen. Aus diesem Grund werden die Prüfvorschriften durch die BSI-TR ergänzt und bieten in diesem Rahmen Kriterien und Methoden für Konformitätsprüfungen an, die im Gegensatz zu den Maßnahmen des IT-Grundschutz-Kompendiums genaue technische Spezifikationen enthalten [49]. In vielen Fällen werden dazu bestehende Standards aus beispielsweise Common Criteria (CC) übernommen oder ergänzt. Bis zum heutigen Tag hat das BSI 63 BSI-TR veröffentlicht, von denen die Folgenden für diese Master-Thesis relevant sind [50]:

- BSI-TR-02102: Kryptographische Verfahren: Empfehlungen und Schlüssellängen
- BSI-TR-02103: X.509-Zertifikate und Zertifizierungspfadvalidierung
- BSI-TR-03111: Elliptische-Kurven-Kryptographie (ECC)

Aus der vorangegangenen Auflistung ist bereits ersichtlich, dass die im BSI-TR enthaltenen technischen Spezifikationen zwar präzise, aber generisch auf fast alle Anwendungsbereiche angewendet werden können. Es werden demnach keine Empfehlungen formuliert, die auf ein bestimmtes Betriebssystem oder eine bestimmte Anwendung mit einer konkreten Version zugeschnitten sind. Ihr maximal möglicher Umsetzungsgrad muss daher durch den Anwender bzw. den IT-Administrator ermittelt werden oder wird durch die individuellen Vorgaben eines ggf. vorhandenen Anforderungsinhabers definiert.

2.2.2 Center for Internet Security Benchmarks und Controls

Das CIS ist eine Non-Profit-Organisation mit Hauptsitz in New York, die im Oktober 2000 gegründet wurde. Auslöser für die Gründung war die zunehmende Vernetzung von Unternehmen bei zugleich fehlenden detaillierten technischen Sicherheitsspezifikationen für den operativen Schutz von Computersystemen [51].

Die Schaffung derartiger Standards erfordert einen Prozess mit dessen Unterstützung eine Vielzahl von Akteuren einen Konsens über sichere Systemkonfigurationen für eine Reihe von Plattformen erzielen können [52]. Dieser Konsensprozess wurde durch den Zusammenschluss von internationalen Wirtschaftsprüfungsverbänden, Nutzerorganisationen und einigen Spezialisten für Informationssicherheit initiiert, die ihr technisches Wissen und ihre finanziellen Ressourcen im CIS zur Entwicklung eben jener Sicherheitsspezifikationen für die am weitesten verbreiteten Betriebssysteme und Anwendungen gebündelt haben [53]. Dadurch wird das Ziel verfolgt, einen nahezu vollständigen Überblick über das Absicherungsziel zu erhalten, mehrere Absicherungsoptionen in Betracht zu ziehen und möglichst keine Härtungsmaßnahmen außer Acht zu lassen. Darüber hinaus ist dieser Ansatz besonders effektiv, da das Fachwissen erfahrener Experten konzentriert und durch ihre freiwillige Teilnahme besonders kosteneffizient genutzt sowie organisationsübergreifend ausgetauscht wird [54].

Zunächst sind daraus die CIS-Benchmarks und ab dem Jahr 2015 auch die CIS-Controls entstanden [51], für deren Zusammenstellung, Aktualisierung und Weiterentwicklung seither das CIS verantwortlich ist. Die CIS-Benchmarks dienen als technische Leitlinien, die für die Absicherung konkreter Betriebssysteme, Anwendungen oder Netzkomponenten angewendet werden können. Im Gegensatz dazu bilden die CIS-Controls einen Katalog von 18 Schlüsselmaßnahmen zur Abwehr oder Abschwächung bzw. Folgenminderung bekannter Angriffe [55], die zur Gewährleistung der IT-Sicherheit einer Organisation im Allgemeinen eingesetzt werden können. Wie auch die CIS-Benchmarks basiert ihre Entwicklung auf dem Konsensprinzip und formuliert Maßnahmen in einer Weise, die einen hohen Automatisierungsgrad zur Durchsetzung der Kontrollen zulassen [56].

2.2.3 Security Technical Implementation Guides

Zur Verbesserung der Cybersicherheit und zur Einhaltung von Vorschriften zur Compliance müssen die US-amerikanische Regierung, Bundesbehörden, zivile Behörden und die Streitkräfte einige Richtlinien für Sicherheits- und Datenschutzkontrollen zum Schutz sensibler Daten berücksichtigen [57]. Angesichts der unterschiedlichen Schutzbedarfe dieser einzelnen Einrichtungen und Institutionen gelten hierzu jedoch mitunter abweichende Mindestanforderungen und Zuständigkeiten. Während das National Institute of Standards and Technology (NIST) Referenzrichtlinien für die gesamte Bundesregierung bereitstellt [58] und der Federal Information Securi-

ty Modernization Act (FISMA) Richtlinien für zivile Behörden enthält, gibt es für Systeme des United States Department of Defense (DoD) noch eine weitere Ebene von Anforderungen, die von der DISA herausgegeben werden [59]. Seit 1998 spielt die DISA durch die Bereitstellung der STIGs eine entscheidende Rolle bei der Verbesserung der Sicherheitslage der Sicherheitssysteme des DoD [60]. Ähnlich wie die CIS-Benchmarks stellen die DISA STIGs technische Leitlinien dar, die zur Absicherung von Informationssystemen/Software verwendet werden können, die andernfalls für einen bösartigen Computerangriff anfälliger wären. Obwohl das DoD speziell auf die CIS-Benchmarks verweist, ist für die Abriegelung von IA-aktivierten Geräten/-Systemen häufig nur die Anwendung der STIGs zulässig. Hierbei handelt es sich um Geräte, deren Hauptaufgabe nicht im Bereich der Sicherheit liegt, die aber als Teil ihrer beabsichtigten Betriebsfunktionen Sicherheitsdienste anbieten (sicherheitsfähige Webbrowser, Screening-Router oder auch vertrauenswürdige Betriebssysteme) [61]. Aus diesem Grund bietet das CIS Betriebssystem-Benchmarks an, die dem entsprechenden STIGs-Standard für Betriebssystemsicherheit zugeordnet sind. Verfügbar sind diese für die Betriebssysteme Red Hat Enterprise Linux 7 & 8, Ubuntu Linux 20.04 LTS, Amazon Linux 2 und Microsoft Windows Server 2016 & 2019, die die Maßnahmen aus CIS und STIGs somit aufeinander abstimmen und vereinen [62, 63]. Mit Stand vom 9. August 2022 hat die DISA etwa 530 STIGs veröffentlicht [64] und gibt in halbjährlichen Abständen weitere heraus. Dabei stellt die DISA sowohl die Anforderungen als auch die Werkzeuge zur Validierung und Umsetzung der Sicherheitsanforderungen zur Verfügung.

2.3 Progress Chef InSpec

Chef war ein US-amerikanisches Unternehmen, das 2008 unter dem Namen Opscode mit Hauptsitz in Seattle gegründet wurde. Die Kernkompetenz des Unternehmens lag in der Entwicklung von Anwendungen für das Konfigurationsmanagement [65]. Im September 2020 gab das Unternehmen bekannt, dass es von Progress Software übernommen wird, mit einem geplanten Übernahmezeitpunkt im Oktober 2020. Das fusionierte Unternehmen wurde anschließend in Progress Chef umbenannt [66]. Bereits vor der Fusion wurde das Portfolio im Laufe der Jahre um mehrere Produkte erweitert, die über das klassische Konfigurationsmanagement hinausgehen und Automatisierungslösungen für Infrastrukturen und Anwendungen umfassen [67].

Progress Chef InSpec ist eine Komponente des Chef Enterprise Automation Stacks und verfügt über eine anwendungsspezifische Sprache zur Beschreibung von Sicher-

heits- und Compliance-Regeln, die von Anwendungsentwicklern, Operations- und Sicherheitsingenieuren gemeinsam genutzt werden kann. Compliance-, Sicherheits- und andere regulatorische Anforderungen werden so in InSpec-Anwendungscode überführt, der auf klassischen Servern, Containern und Cloud-APIs ausgeführt werden kann [68]. Durch dessen automatisierte, wiederkehrende Ausführung kann die Durchsetzung kohärenter Standards in jeder verwalteten Umgebung sowie in jedem Entwicklungsstadium in wenigen Minuten auditiert und so dauerhaft sichergestellt werden [69].

Der InSpec-Anwendungscode wird in einem Profil zusammengeführt, um Kontrollen zu organisieren und die Verwaltung von Abhängigkeiten sowie die Wiederverwendung von Code zu unterstützen. Jedes Profil wird üblicherweise auf der Grundlage der repräsentierten Sicherheits- und Compliance-Regeln ausgegliedert und stellt somit eine eigenständige Struktur mit eigenem Distributions- und Ausführungsfluss dar. Dies kann zum Beispiel ein vollständiges Betriebssystem, eine Anwendung oder die Dienste einer Cloud-Computing-Plattform umfassen. Es ist jedoch auch möglich, mehrere Profile miteinander zu verbinden [70]. Ein typischer Profil-Verzeichnisbaum wird nachstehend am Beispiel der MySQL-Baseline des auf InSpec basierenden DevSec Hardening Frameworks veranschaulicht [71]:

```
/ ........................ Wurzelverzeichnis der MySQL-Baseline (Beispiel)
├── controls ............................ Speicherort aller InSpec-Kontrollen
│   ├── mysql_conf.rb ..................... MySQL-Konfigurationskontrollen
│   └── mysql_db.rb ........................ MySQL-Datenbankkontrollen
├── libraries ............ Speicherort aller InSpec-Ressourcenerweiterungen
│   ├── mysql_distribution.rb ............ MySQL-Plattformidentifizierung
│   └── mysql_version.rb ..................... MySQL-Versionsfeststellung
├── files ............................ Speicherort aller zusätzlichen Dateien
│   └── my.cnf ............................... MySQL-Konfigurationsdatei
├── README.md ............................ Beschreibung des InSpec-Profils
└── inspec.yml ........................... Konfiguration des InSpec-Profils
```

Jedes Profil muss über eine `inspec.yml`-Datei verfügen, die den Einstiegspunkt für die Programmausführung darstellt und beschreibende Informationen über das Profil sowie die erforderlichen Abhängigkeiten enthält. Zu Letzteren gehören unter anderem die erforderliche Mindestversion von InSpec, eine Liste der unterstützten Plattformnamen oder -familien, eine Auflistung aller abhängigen Drittanbieterprofile und alle zusätzlichen Programmabhängigkeiten in Form von Gem-Paketen [70].

Als Beispiel für einen Plattformnamen kann Microsoft Windows Server 2022, für eine Plattformfamilie alle Red Hat Enterprise Linux-Derivate, für ein abhängiges Drittanbieterprofil die DevSec Linux-Baseline und für ein notwendiges Gem-Paket der MySQL-Kommandozeilen-Client herangezogen werden.

Ebenfalls erforderlich ist ein `controls`-Ordner, der den InSpec-Anwendungscode in Form einer Ruby-Domain Specific Language (DSL) enthält. Die Ruby-DSL wird von InSpec bereitgestellt und enthält eine Sammlung von Schnittstellen und Ressourcen, die eine schnelle und leicht verständliche Programmierung von Sicherheits- und Compliance-Regeln ermöglichen [70, 72]. Dies wird in Listing 3 am Beispiel der Prüfung der Eigentums- und Zugriffsrechte der MySQL-Protokolldatei veranschaulicht. Zunächst wird in den Zeilen 2 - 7 die richtige Eigentümergruppe auf Basis der vorhandenen Plattformfamilie ermittelt. Variationen bei den Dateiberechtigungen, der Konfigurationsstruktur oder sogar dem Namen der Anwendungen selbst auf Grundlage des Plattformnamens oder der -familie müssen nicht selten berücksichtigt werden. Insbesondere bei älteren, schon lange entwickelten Anwendungen sind aus der Historie heraus abweichende interne Bezeichnungen für die gleiche Anwendung entstanden. Ein Beispiel dafür ist der Apache-HTTP-Server, der unter Debian-Derivaten im Gegensatz zu fast allen anderen Linux-Distributionen als apache2 anstelle von httpd betrieben wird [73]. In den Zeilen 13 und 14 werden anschließend die entsprechenden Ruby-DSL-Ressourcen verwendet, um den korrekten Eigentümer sowie die korrekte Eigentümergruppe in Abhängigkeit zur Plattformfamilie zu verifizieren. Ferner wird in Zeile 15 bis 17 gewährleistet, dass die MySQL-Protokolldatei nicht für weitere Benutzer zugänglich ist [74].

```
1  # OS-abhängige Variablen setzen
2  case os[:family]
3  when 'ubuntu', 'debian'
4    mysql_log_group = 'adm'
5  else
6    mysql_log_group = 'mysql'
7  end
8
9  control 'oracle-mysql-db-36' do
10   impact 0.5
11   title '3.6 Ensure general_log_file Has Appropriate Permissions
     ↪  (Automated)'
12   describe file(mysql_log_file) do
13     it { should be_owned_by 'mysql' }
14     it { should be_grouped_into mysql_log_group }
```

```
15      it { should_not be_readable.by('others') }
16      it { should_not be_writable.by('others') }
17      it { should_not be_executable.by('others') }
18    end
19  end
```

Listing 3: Auditierung von Konfigurationsparametern nach CIS Oracle MySQL Enterprise Edition 8.0 Benchmark v1.2.0 für Control 3.6: „Sicherstellung, dass general_log_file die angemessenen Berechtigungen aufweist“

Neben diesen sehr trivialen Prüfungen, die für verschiedene Anwendungen und Dienste in identischer Weise durchgeführt werden können, ist es auch notwendig, die Ausführung anwendungsspezifischer Kontrollen zu ermöglichen [75]. Bei anwendungsspezifischen Kontrollen handelt es sich um Prüfungen, für die keine generische Ruby-DSL-Ressource von InSpec zur Verfügung steht. Das Listing 4 zeigt die Implementierung einer anwendungsspezifischen Kontrolle zur Überprüfung der Hostnamen aller im DBMS angelegten Benutzer. Zu diesem Zweck wird in Zeile 4 der MySQL-Befehlszeilen-Client verwendet, um eine SQL-Abfrage direkt auf der Datenbank auszuführen. Die Ausgabe bzw. das Ergebnis der Abfrage wird in Zeile 5 wiederum mit einer DSL-Ressource unter Zuhilfenahme eines regulären Ausdrucks evaluiert.

```
1  control 'oracle-mysql-db-76' do
2    impact 0.5
3    title '7.6 Ensure No Users Have Wildcard Hostnames (Automated)'
4    describe command("mysql -u#{user} -p#{pass} -sN -e 'select count(*) from
     ↪  mysql.user where host=\"%\"'") do
5      its(:stdout) { should match(/^0/) }
6    end
7  end
```

Listing 4: Auditierung von Konfigurationsparametern nach CIS Oracle MySQL Enterprise Edition 8.0 Benchmark v1.2.0 für Control 7.6: „Sicherstellung, dass keine Benutzer Wildcard-Hostnamen aufweisen“

Optional ist hingegen die Verwendung des `libraries`-Ordners, welcher verwendet wird, um eigene InSpec-Ressourcenerweiterungen zu implementieren, die sich für eine Wiederverwendung eignen [70, 76]. Ein Beispiel hierfür sind die Zeilen 2 bis 7 aus dem zuvor vorgestellten Listing 3. Die Bestimmung der zugrundeliegenden Platt-

formfamilie ist nicht nur für die Prüfung der Eigentümergruppe relevant, sondern wird auch für einige weitere Kontrollen benötigt. Daher ist es sinnvoll, entsprechende Abhängigkeiten auszulagern und deren Ergebnis vor der Ausführung aller Kontrollen zur Optimierung des Programmablaufs zwischenzuspeichern.

Ebenfalls optional ist der Ordner `files`, der zusätzliche Dateien enthält, auf die ein Profil zugreifen kann. Dies können testrelevante Abhängigkeiten sein, bei denen es sich nicht um InSpec-Ressourcenerweiterungen handelt, oder auch gehärtete Beispielkonfigurationen für den Anwender, wie auch weitere Zusatzinformationen. Häufig wird jedoch auch die `README.md` für diesen Zweck verwendet [70].

3 Entwicklung des Vorgehensmodells

In diesem Kapitel wird die im Rahmen dieser Master-Thesis angewendete Vorgehensweise zur Erstellung eines neuen Vorgehensmodells zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme beschrieben. Die Grundlage dafür stellen die in Abschnitt 2.2 vorgestellten Konfigurations- und Compliance-Richtlinien dar. Hierzu werden einige der BSI-Bausteine und ausgewählte CIS-Benchmarks sowie STIGs-Leitfäden, die im Zusammenhang mit nicht-relationalen Datenbanksystemen anwendbar sind, untersucht. Im Anschluss an die Untersuchung werden die als anwendbar identifizierten Maßnahmen aus allen betrachteten Standards in einer neuen Struktur vereint, indem Überschneidungen zusammengeführt und dadurch alle Inhalte in einem neuen Vorgehensmodell gebündelt werden.

Die Reduzierung der Auswertung auf eine begrenzte Auswahl aus den einzelnen Standards ergibt sich aus der Notwendigkeit, den Gesamtaufwand in ein vertretbares Verhältnis zu dem angestrebten Erkenntnisgewinn zu setzen. Außerdem ist im Zusammenhang mit der Untersuchung zu beachten, dass sich der Geltungsbereich dieser Master-Thesis grundsätzlich auf technische Maßnahmen beschränkt, während organisatorische und prozessuale Aspekte nicht berücksichtigt werden. Die Auswirkungen der Härtungsmaßnahmen auf die Performance des jeweiligen Datenbanksystems wird ebenfalls nicht untersucht.

3.1 Auswahl der BSI-Bausteine, CIS-Benchmarks und STIGs-Leitfäden

Bei der Betrachtung der in Abschnitt 2.2 vorgestellten Werkzeuge für die Informationssicherheit wurde die Rolle des IT-Grundschutz-Kompendiums erörtert, dessen Kern die IT-Grundschutz-Bausteine bilden. Auf einer übergeordneten Ebene werden diese in prozessorientierte und systemorientierte Bausteine unterteilt und nach verwandten Themen eingeordnet [77]. Den Kern eines jeden Bausteins bilden die Sicherheitsanforderungen, die eine oder mehrere Maßnahmen enthalten, die zur Erfüllung der Anforderung umgesetzt werden müssen. Für die Entwicklung des Vorgehensmodells wurden die Maßnahmen sämtlicher Sicherheitsanforderungen von sowohl allen

34 prozessorientierten als auch allen 70 systemorientierten Bausteinen auf ihre Anwendbarkeit hin untersucht. Im Zuge dessen konnten Anforderungen mit relevanten Maßnahmen in den folgenden Bausteinen identifiziert werden [78]:

ORP.4 Identitäts- und Berechtigungsmanagement	32 Maßnahmen
CON.1 Kryptokonzept	13 Maßnahmen
CON.8 Software-Entwicklung	9 Maßnahmen
CON.10 Entwicklung von Webanwendungen	2 Maßnahmen
OPS.1.1.3 Patch- und Änderungsmanagement	3 Maßnahmen
OPS.1.1.4 Schutz vor Schadprogrammen	1 Maßnahme
OPS.1.1.5 Protokollierung	7 Maßnahmen
APP.4.3 Relationale Datenbanksysteme	10 Maßnahmen
APP.6 Allgemeine Software	5 Maßnahmen

Alle Maßnahmen wurden mit einer fortlaufenden Nummer mit dem Präfix „B“ versehen, um nachfolgend eine Gruppierung und Referenzierung zu ermöglichen. Alle Maßnahmen, die nicht berücksichtigt werden, enthalten eine entsprechende Begründung, zum Beispiel weil diese nicht in Form von technischen Maßnahmen umgesetzt werden können und somit außerhalb des Geltungsbereichs des Vorgehensmodells liegen. Eine vollständige Übersicht dieser kann in Anhang F eingesehen werden.

Neben den BSI IT-Grundschutz-Bausteinen wurden die folgenden drei CIS-Benchmarks in die Erstellung des Vorgehensmodells einbezogen [79, 80, 81]:

CIS Apache Cassandra 3.11 Benchmark	16 Maßnahmen
CIS MongoDB 5 Benchmark	22 Maßnahmen
CIS PostgreSQL 14 Benchmark	47 Maßnahmen

Wie zuvor wurden alle Maßnahmen mit einer fortlaufenden Nummer, an dieser Stelle mit dem Präfix „C“, versehen oder es wurde eine entsprechende Begründung im Falle einer Nichtberücksichtigung gegeben. Eine vollständige Übersicht dieser kann in Anhang G eingesehen werden.

Abschließend wurden die folgenden zwei STIGs-Leitfäden berücksichtigt [82, 83]:

MongoDB Enterprise Advanced 4.x STIG V1, R1	45 Maßnahmen
Redis Enterprise 6.x STIG V1, R1	70 Maßnahmen

Wie zuvor wurden alle Maßnahmen mit einer fortlaufenden Nummer, hier mit dem Präfix „S“, versehen oder es wurde eine entsprechende Begründung im Falle einer Nichtberücksichtigung gegeben. Eine vollständige Übersicht dieser kann in Anhang H eingesehen werden.

3.2 Umsetzung des Vorgehensmodells

In diesem Abschnitt werden die ermittelten Maßnahmen aus den BSI-Bausteinen und den CIS- sowie STIGs-Leitfäden zu neuen Einzelanforderungen im Sinne eines neuen Vorgehensmodells/einer neuen Basisrichtline für die Absicherung nichtrelationaler Datenbanksysteme zusammengefasst. Diese werden ebenfalls mit einer fortlaufenden Nummer mit dem Präfix „A“ versehen. Für jede abgeleitete Anforderung wird im Folgenden eine Beschreibung des beabsichtigten Verwendungszwecks und, wo möglich, Hinweise zur Umsetzung gegeben. Darüber hinaus werden diese anschließend durch die Empfehlungen aus den BSI-TR, weiteren Best-Practice-Ansätzen sowie Herstellerangaben und -Leitlinien ergänzt.

Von besonderer Bedeutung ist die Anforderung A1, die im Vorgehensmodell als Sammelbecken für die Implementierung eines Demingkreises dient. Alle technischen Maßnahmen, die im Rahmen der Anwendung auf ein Datenbanksystem im folgenden Kapitel 4 identifiziert werden und sich keiner der Anforderungen zweifelsfrei zuordnen lassen, werden dieser Anforderung zugeordnet. Die so erfassten Maßnahmen können anschließend als Grundlage für die Fortschreibung/Weiterentwicklung des Vorgehensmodells genutzt werden, indem zusätzliche, neue Anforderungen formuliert werden und auf diese Weise das Modell erweitert wird.

3.2.1 Grundprinzipien des Vorgehensmodells

A1 Die Standardwerte aller sicherheitsrelevanten Konfigurationsparameter müssen explizit festgelegt werden.

> **BSI:** B47, B80 | **STIGs:** S45, S76

Beschreibung Zur Identifikation aller sicherheitsrelevanten Konfigurationsparameter ist grundsätzlich ein holistischer Ansatz zu verfolgen. Die vorliegenden Anforderungen, die auf einer komprimierten Zusammenfassung von 282 Einzelanforderungen aus 6 verschiedenen Leitfäden beruhen, decken nahezu alle Maßnahmen ab, die aus technischer Sicht im Zusammenhang mit dem sicheren Betrieb von nicht-relationalen Datenbanksystemen zu berücksichtigen sind. Dennoch müssen weiterhin sämtliche Konfigurationsparameter über die hier vorgenommene Klassifizierung hinaus auf ihre sicherheitstechnische Relevanz hin analysiert und gegebenenfalls mit einem sicheren Konfigurationswert versehen werden. Dies liegt zum einen an der großen Vielfalt nicht-relationaler Datenbanksysteme, die zum Teil auf die Ausschlussdefinition des Begriffs selbst zurückzuführen ist, und zum anderen an der hohen Geschwindigkeit der Weiterentwicklung, sodass eine Abdeckung aller sicherheitsrelevanten Aspekte in einem Vorgehensmodell nicht abschließend sichergestellt werden kann. Dies ist insbesondere auch dann von Bedeutung, wenn der Standardwert des jeweiligen Konfigurationsparameters bereits einer sicheren Konfiguration entspricht, da sich Standardwerte durch die Installation von Updates nachträglich negativ verändern können.

Umsetzung Die Dokumentation des DBMS ist auf sicherheitsrelevante Konfigurationsparameter zu untersuchen. Außerdem sollen die Inhalte der Standard-Konfiguration und die Empfehlungen aus Best-Practice-Ansätzen bei der Untersuchung berücksichtigt werden, da insbesondere bei Open-Source-Projekten neue Parameter und Funktionalitäten mitunter erst verzögert in der Dokumentation beschrieben werden.

A2 Nicht benötigte Plug-ins/Software-Erweiterungen und Funktionen müssen deinstalliert oder deaktiviert werden.

BSI: B80, B81, B82 | **STIGs:** S4, S42, S77, S79, S80, S82, S112

Beschreibung In den meisten Fällen entspricht die Standard-Konfiguration eines Datenbanksystems einem Zustand, der dem Benutzer den größtmöglichen Funktionsumfang präsentiert. Alle Netzwerkfunktionen, Ports, Protokolle, Dienste und Erweiterungen sind aktiviert und die Einstellungen entsprechen einem Betriebszustand, der den niedrigschwelligsten Einstieg in die Nutzung der Anwendung ermöglicht. In den wenigsten Fällen wird jedoch der volle Funktionsumfang im produktiven Betrieb ausgeschöpft. Aus Sicht der Systemhärtung wird es einem Angreifer dadurch massiv erleichtert, das Da-

tenbanksystem zu kompromittieren und damit für weitere Angriffe oder Datenabflüsse zu nutzen. Nicht benötigte Plug-ins/Software-Erweiterungen und Funktionen sollten daher zur Reduzierung der Angriffsfläche deaktiviert oder, falls möglich, deinstalliert werden.

Umsetzung Aufgrund der unterschiedlichen Anforderungen, die direkt mit der Zielsetzung und dem Einsatzkontext des Datenbanksystems zusammenhängen, ist es nicht möglich, einen allgemeingültigen Ansatz zur Umsetzung darzustellen. Mit Blick auf die spätere Anwendung des Vorgehensmodells kann jedoch das Ziel formuliert werden, das Datenbanksystem auf seine Kernfunktionalität zu reduzieren, um eine möglichst große Anzahl von Härtungsmaßnahmen ableiten zu können. Begründete Abweichungen zur Bewältigung des Einsatzkontextes müssen daher bei der Umsetzung berücksichtigt und dokumentiert werden.

3.2.2 Installation und Updates

A3 Der Aktualisierungsmechanismus muss sicher konfiguriert werden.

BSI: B57 | **STIGs:** S16, S73

Beschreibung Die zeitnahe Installation von Patches ist im Rahmen des Lebenszyklus des Datenbanksystems entscheidend. Dabei muss sichergestellt werden, dass der Update-Mechanismus gesichert und geeignet konfiguriert ist.

Umsetzung Das DBMS ist daraufhin zu untersuchen, welche Aktualisierungsmechanismen zur Verfügung stehen. Sollten mehrere Verfahren zur Auswahl stehen, muss der Sicherste gewählt werden. Darüber hinaus muss sichergestellt werden, dass der Zugriff auf den Aktualisierungsmechanismus nur für autorisierte Benutzer möglich ist und somit nicht von einem Angreifer dazu missbraucht werden kann, kompromittierte Aktualisierungspakete einzuspielen.

A4 Die Herkunft der Software-Installations- und -Aktualisierungspakete aus vertrauenswürdigen Quellen muss gewährleistet werden.

BSI: B59, B78 | **CIS:** C39

Beschreibung Für die Installation oder Aktualisierung von Anwendungen

steht unter Linux üblicherweise ein Paketmanager zur Verfügung, der sich je nach Distribution unterscheidet. Weit verbreitet sind die Paketverwaltungssysteme

- Debian Package Management System (DPKG) unter Debian-basierten Derivaten mit dem Advanced Packaging Tool (APT) oder dem Aptitude Package Manager als Frontend,
- Red Hat Package Manager (RPM) unter Red Hat Enterprise Linux oder Fedora mit Yellowdog Updater, Modified (YUM) oder Dandified Yum (DNF) als Frontend,
- Pacman Package Manager unter Arch Linux oder auch
- Zypper Package Manager unter openSUSE.

Umsetzung Die Konfiguration des Paketverwaltungssystems des zugrundeliegenden Betriebssystems muss überprüft werden, um festzustellen, ob die konfigurierten Paketquellen vertrauenswürdig sind. In der Regel werden die offiziellen Paketquellen des Betriebssystems, die des Entwicklers des Datenbanksystems oder ein eigener unternehmensinterner Update-Server anerkannt. Die Installation des Datenbanksystems ohne ein Paketverwaltungssystem, z. B. durch die lokale Kompilierung der Anwendung bei Open-Source-Projekten, ist nicht zulässig.

A5 Die Herkunft von Plug-ins/Software-Erweiterungen aus vertrauenswürdigen Quellen muss gewährleistet werden.

BSI: B53

Beschreibung Der Funktionsumfang kann in vielen Fällen durch die Installation von Plug-ins/Software-Erweiterungen erweitert werden. Dabei kann zwischen Erweiterungen des Entwicklers selbst und von Drittanbietern unterschieden werden. Insbesondere Letztere stellen ein erhebliches Risiko für das Datenbanksystem dar, da zumeist weder die Qualität des Programmcodes noch die Authentizität der Erweiterung sichergestellt werden kann.

Umsetzung Das DBMS muss hinsichtlich der Möglichkeit der Installation von Plug-ins/Software-Erweiterungen untersucht werden. Derartige Erweiterungen, die über das Paketverwaltungssystem nachinstalliert werden können, sind hinsichtlich ihrer Vertraulichkeit bzw. Authentizität in der Regel unbedenklich, müssen aber hinsichtlich ihres Funktionsumfangs unter Sicherheits-

aspekten betrachtet werden. Ist die Installation von Erweiterungen hingegen direkt in die Anwendung integriert, zum Beispiel in Form eines herstellereigenen Online-Marktplatzes, sollte diese Funktion deaktiviert oder der Zugang gesperrt werden. Kann das angestrebte Einsatzziel jedoch nicht so (um)gestaltet werden kann, dass der Einsatz von Software-Erweiterungen nicht erforderlich ist, sollten nur die offiziellen Erweiterungen des Herstellers verwendet werden.

A6 Die Integrität der Software-Installations- und -Aktualisierungspakete muss verifiziert werden.

BSI: B58, B79

Beschreibung Um die Integrität von Software-Installations- und Aktualisierungspaketen zu gewährleisten, verwenden Paketverwaltungssysteme Prüfsummen und digitale Signaturen. Diese werden zusammen mit den Softwarepaketen auf dem Aktualisierungsserver gespeichert und können vor der Installation lokal validiert werden.

Umsetzung Es muss sichergestellt werden, dass der Keyring für das jeweilige Paketverwaltungssystem korrekt konfiguriert ist und nur gültige GPG-Schlüssel aus den offiziellen Paketquellen des Betriebssystems enthält. Insbesondere bei der Verwendung von APT muss sichergestellt werden, dass Signaturen aus einem Drittanbieter-Repository nur für Pakete akzeptiert werden, die aus diesem heraus installiert wurden. Es darf kein GPG-Cross-Signing betrieben werden.

A7 Die Integrität der Plug-ins/Software-Erweiterungen muss verifiziert werden.

BSI: B54

Beschreibung Die Integrität der Plug-ins/Software-Erweiterungen muss ebenfalls gewährleistet werden.

Umsetzung Werden die Plug-ins/Software-Erweiterungen über das Paketverwaltungssystem des Betriebssystems bezogen, so ist die Umsetzung mit der vorherigen Anforderung A6 identisch. Wird jedoch entgegen der Empfehlung aus Anforderung A5 ein integrierter Online-Marktplatz für diesen Zweck genutzt, so muss die Gewährleistung der Integrität gesondert betrachtet werden.

Ursächlich hierfür ist, dass in diesem Fall der Entwickler des Datenbanksystems für die Bereitstellung eines Mechanismus für die Integritätsprüfung verantwortlich ist. Daher muss bei der Nutzung eines anwendungseigenen Online-Marktplatzes sichergestellt werden, dass ein den aktuellen Sicherheitsanforderungen genügender Standard für die Integritätsprüfung implementiert und dieser möglichst sicher konfiguriert ist.

A8 Die Version der Software und der Plug-ins/Software-Erweiterungen müssen vom Hersteller unterstützt werden.

BSI: B80 | **STIGs:** S44, S78

Beschreibung Das Datenbanksystem und die dazugehörigen Plug-ins/Software-Erweiterungen müssen eine vom Hersteller unterstützte Version aufweisen. Eine fehlende Unterstützung birgt potentiell Sicherheitsrisiken, da neu entdeckte Schwachstellen und Anwendungsfehler in der Regel nicht mehr beseitigt werden. Aus diesem Grund darf eine Anwendung bzw. entsprechende Software-Erweiterung ab diesem Zeitpunkt nicht mehr verwendet werden und muss auf eine unterstützte Version aktualisiert werden.

Umsetzung Die Version des Datenbanksystems und aller aktivierten Software-Erweiterungen muss abgefragt und mit einer Liste der unterstützten Versionen verglichen werden.

A9 Die Installation von Aktualisierungspaketen muss zeitnah erfolgen.

STIGs: S43, S113

Beschreibung Trotz der zunehmenden Standardisierung von Entwicklungsprozessen, dem Einsatz von Static Application Security Testing (SAST)-Tools sowie DevSecOps werden nach wie vor täglich neue Sicherheitslücken in Anwendungen aufgedeckt und veröffentlicht. Die unverzügliche Installation von Aktualisierungspaketen ist daher aufgrund ihrer meist sicherheitsrelevanten Bedeutung zwingend erforderlich.

Umsetzung Die Version des Datenbanksystems und aller aktivierten Software-Erweiterungen muss ermittelt und mit der aktuellsten verfügbaren Version abgeglichen werden. Stimmen diese nicht überein, muss das entsprechende Aktualisierungspaket umgehend angewendet werden.

3.2.3 Authentifizierung

A10 Die Authentifizierung muss konfiguriert und aktiv sein.

BSI: B10 | **CIS:** C4, C17 | **STIGs:** S6, S20, S84, S91

Beschreibung Der Zugriff auf das DBMS und alle zugehörigen Dienste muss durch eine wirksame Authentifizierung und damit Identifizierung der zugreifenden Benutzer, Dienste oder IT-Systeme geschützt werden. Dies dient dem Ausschluss unberechtigter Zugriffe durch nicht autorisierte Entitäten. Ohne erfolgreiche Authentifizierung darf ein Zugriff niemals erfolgen.

Umsetzung Das DBMS sollte im Hinblick auf die Konfiguration und Aktivierung einer Authentifizierungsfunktion zur Beschränkung des Zugriffs überprüft werden.

A11 Sind mehrere Authentifizierungsmechanismen verfügbar, ist das sicherste Verfahren zu verwenden.

BSI: B26, B50 | **STIGs:** S21

Beschreibung Für die Implementierung einer Authentifizierung stehen im Regelfall mehrere Verfahren zur Verfügung, die sich in erster Linie durch die folgenden Eigenschaften unterscheiden:

1. Sicherheit: Was sind mögliche Angriffsvektoren und welche Hürden muss ein Angreifer nehmen, um die Authentifizierung zu umgehen?
2. Implementierbarkeit: Wie komplex ist eine systemweite Implementierung unter Berücksichtigung der Authentifizierung von sowohl Personen als auch Diensten?
3. Nutzerfreundlichkeit: Wie intuitiv und komfortabel kann der Authentifizierungsprozess gestaltet werden?

Die aufgeführten Eigenschaften sollten entsprechend ihrer Nennung priorisiert werden, wobei hinsichtlich der Implementierbarkeit ein Authentifizierungsmechanismus durch ein externes System, wie z. B. ein Identity and Access Management (IAM)-System, gegenüber einer lokalen Authentifizierung zu bevorzugen ist.

Umsetzung Für das DBMS muss untersucht werden, welche Authentifizie-

rungsmethoden zur Verfügung stehen. Im Allgemeinen sollte die Authentifizierung mit Kerberos oder Lightweight Directory Access Protocol (LDAP) über ein IAM-System oder mittels Zertifikaten erfolgen. Die Verwendung von Passwörtern sollte vermieden werden.

A12 Die Authentifizierung aller Teilnehmer in einem Cluster muss konfiguriert und aktiv sein.

CIS: C19

Beschreibung Einige nicht-relationale DBMS unterstützen die Konfiguration eines Clusters, um die Anzahl der möglichen Lese- und Schreiboperationen horizontal über mehrere Server hinweg zu skalieren. Außerdem wird dadurch die Hochverfügbarkeit und damit die Zuverlässigkeit der Datensätze verbessert. Für die Umsetzung ist eine Synchronisation bzw. Replikation der Daten zwischen allen Teilnehmern des Clusters notwendig. Dieser Prozess muss vor unberechtigtem Zugriff geschützt werden.

Umsetzung Die vom DBMS für den Betrieb des Clusters zur Verfügung stehenden Authentifizierungsmethoden müssen identifiziert und ein Verfahren aktiviert werden. Im Gegensatz zu Anforderung A11 stehen an dieser Stelle nur die Verfahren zur Auswahl, die eine Automatisierung des Authentifizierungsprozesses ermöglichen.

A13 Die Authentifizierung muss an sämtlichen Schnittstellen/Interfaces erfolgen.

CIS: C18, C76, C77

Beschreibung Einige Datenbanksysteme ermöglichen eine differenzierte Konfiguration der Authentifizierung, z. B. auf der Grundlage der für den Zugriff verwendeten Schnittstelle oder der IP-Adresse des Absenders. In diesem Zusammenhang erlauben einige Standard-Konfigurationen den Zugriff auf das Datenbanksystem ohne Authentifizierung, beispielsweise wenn der Zugriff über den lokalen Host erfolgt. Dies kann es einem Angreifer ermöglichen, die Authentifizierung zu umgehen. Darüber hinaus kann die Rückverfolgbarkeit jeder Datenbankaktivität zu einem bestimmten Benutzer nicht gewährleistet werden.

Umsetzung Die Authentifizierung muss so konfiguriert sein, dass sie immer

erzwungen wird. Es dürfen keine Ausnahmeregelungen für die Authentifizierung gewährt werden, unabhängig davon, von welcher Schnittstelle aus der Zugang erfolgt oder auf welche Datenbankfunktionen zugegriffen werden soll.

A14 Die Authentifizierung muss, wenn möglich, mehrere Authentifizierungsmerkmale umfassen (Multi-Faktor-Authentifizierung).

BSI: B7, B22, B32

Beschreibung Die Authentifizierung anhand mehrerer Faktoren reduziert die Wahrscheinlichkeit eines erfolgreichen unbefugten Zugriffs. Die Implementierung ist insbesondere für die Benutzer-Authentifizierung an den Management-Schnittstellen relevant, während eine Realisierung für die automatisierte Authentifizierung von Diensten oder gar Cluster-Teilnehmern meist nicht möglich bzw. nicht sinnvoll ist.

Umsetzung Das DBMS ist auf die Möglichkeit der Implementierung einer Multi-Faktor-Authentifizierung zu untersuchen. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt durch das DBMS bereitgestellt, sondern durch die Anbindung eines IAM-Systems realisiert.

A15 Sitzungskennungen müssen zufällig erzeugt werden und dürfen kein vorhersehbares Schema aufweisen.

STIGs: S98, S99

Beschreibung Zur Aufrechterhaltung der Authentifizierung werden je nach Methode und verwendeter Schnittstelle unterschiedliche Arten von Session-IDs verwendet. Werden diese nicht hinreichend zufällig und mit ausreichender Länge vergeben bzw. generiert, besteht die Möglichkeit, dass ein Angreifer diese errät oder durch einen Brute-Force-Angriff selbst erzeugt. Im Erfolgsfall kann dies zu einem unbefugten Zugriff etwa auf die Management-Schnittstelle und folglich zur Kompromittierung des Datenbanksystems führen. Aus diesem Grund dürfen Sitzungs-IDs niemals vorhersehbar sein oder sich auf einen konkreten Wertebereich eingrenzen lassen.

Umsetzung Das DBMS muss auf die Konfigurationsmöglichkeit der Session-IDs untersucht werden. Falls möglich und erforderlich, muss die Länge der Sitzungs-IDs entsprechend erhöht und ein sicheres Verfahren mit mehreren Entropie-Quellen für ihre Erzeugung gewählt werden.

A16 Fehlgeschlagene Authentifizierungen dürfen nicht zur Durchführung von Angriffen interpretiert werden können.

BSI: B21 | **STIGs:** S24, S89

Beschreibung Eine Authentifizierung kann aus verschiedenen Gründen fehlschlagen, z. B. wenn das Benutzerkonto nicht existiert, die Benutzerkennung oder das zugehörige Kennwort falsch ist sowie wenn das Benutzerkonto gesperrt bzw. deaktiviert ist. Das Datenbanksystem muss einen ungültigen Authentifizierungsversuch unabhängig vom Grund immer mit einer einheitlichen generischen Antwort zurückweisen, wie etwa „Anmeldung fehlgeschlagen: Ungültige Benutzerkennung oder ungültiges Passwort“. Dadurch soll einem Angreifer kein Interpretationsspielraum hinsichtlich der Ablehnungsgründe gegeben werden, unter anderem um die Durchführung von User-Enumeration-Angriffen zu verhindern oder zumindest zu erschweren. Im Zusammenhang mit der Anmeldung an HTTP-basierten Management-Schnittstellen ist zu beachten, dass nicht nur die für den Benutzer sichtbare Antwort im HTTP-Body entsprechend generisch implementiert werden muss, sondern auch die HTTP-Statuscodes entsprechend angepasst werden müssen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Rückmeldungen im Rahmen der Authentifizierung die oben genannten Kriterien erfüllen oder entsprechend angepasst werden können. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A17 Wenn mehrere Authentifizierungsversuche fehlschlagen, müssen Trigger definiert werden, um weitere Versuche zu verzögern.

BSI: B28

Beschreibung Wenn ein Angreifer direkt mit dem Datenbanksystem bzw. seiner Management-Schnittstelle kommunizieren kann, besteht eine mögliche Vorgehensweise zur Kompromittierung des Systems darin, das Passwort für den administrativen Zugang zu bestimmen. Dazu werden bei einem Brute-Force-Angriff systematisch und automatisiert alle Kombinationen von Zahlen, Buchstaben oder auch Symbolen getestet, bis eine gültige Kombination (das Passwort) ermittelt wurde. Eine effizientere Alternative zu dieser Methode ist

der Wörterbuchangriff, da Benutzer oftmals reale Wort-Zahlen-Kombinationen gegenüber völlig zufälligen Zeichenketten bevorzugen. Um diese Art von Angriff zu schwächen, können verschiedene Maßnahmen ergriffen werden, z. B. die zeitliche Verzögerung eines Authentifizierungsversuchs. Dies kann entweder durch eine Begrenzung der Anzahl möglicher Authentifizierungen pro Zeiteinheit oder durch eine bewusste Verlängerung des Prozesses zur Validierung des Passworts realisiert werden. Diese Art von Schutz ist jedoch nur dann wirksam, wenn er auf der Grundlage des attackierten Benutzerkontos und nicht ausgehend von der IP-Adresse des Absenders implementiert wird, da Letztere durch die Verwendung von anonymen Proxy-Servern leicht umgangen werden kann.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine Zeitverzögerung für Authentifizierungsversuche konfiguriert werden kann. Ein guter Wert, der einerseits die Durchführung eines Brute-Force-Angriffs erheblich beeinträchtigt und andererseits eine akzeptable Auswirkung auf die Benutzererfahrung darstellt, ist eine Authentifizierungszeit von mindestens 5 Sekunden nach bis zu 3 fehlgeschlagenen Authentifizierungsversuchen. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A18 Wenn mehrere Authentifizierungsversuche fehlschlagen, müssen Trigger definiert werden, um aktive Sitzungen zu beenden oder Benutzer zu sperren.

BSI: B30 | **STIGs:** S30, S36, S83

Beschreibung Als Ergänzung oder Alternative zur Anforderung A17 können aktive Sitzungen automatisch beendet oder angegriffene Benutzerkonten gesperrt werden. Es ist jedoch zu beachten, dass solche Maßnahmen von einem Angreifer missbraucht werden können, um durch einen Brute-Force-Angriff auf eine große Anzahl von Konten einen Denial-of-Service-Zustand zu verursachen. Darüber hinaus kann ein Angreifer anhand der Rückmeldungen (vgl. Anforderung A16) oder aus einer Zeitverzögerung (vgl. Anforderung A17) gültige Benutzerkonten ermitteln.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine automatisierte Beendigung von aktiven Sitzungen bzw. eine Sperrung von Benutzer-

konten im Falle eines Angriffs konfiguriert werden kann. Dabei ist zu beachten, dass Konten, die für die Verwaltung oder den fortlaufenden Betrieb der angeschlossenen Anwendungen verwendet werden, von dieser Maßnahme ausgeschlossen sind, um einen Denial-of-Service-Zustand auszuschließen. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A19 Die Gesamtdauer eines Anmeldeversuchs muss begrenzt werden.

BSI: B29

Beschreibung Im Zuge einer Authentifizierung wird eine Verbindung zur entsprechenden Datenbankschnittstelle aufgebaut. Je nach dem dafür verwendeten Protokoll wird eine Verbindung zwischen dem Client und dem Datenbankserver für die Dauer einer noch nicht abgeschlossenen Authentifizierung, d. h. für die Zeit der Eingabe und Übermittlung des Benutzernamens und des Passworts, aufrechterhalten. Dies verbraucht und blockiert Systemressourcen, was von einem Angreifer ausgenutzt werden kann, um einen Denial-of-Service-Zustand zu erreichen. Die Begrenzung der Gesamtdauer eines Anmeldeversuchs auf einen angemessenen Zeitraum stellt sicher, dass der Dienst ständig für den Zugriff verfügbar ist.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Gesamtdauer eines Anmeldeversuchs begrenzt werden kann. Ein guter Wert, der einerseits das Herbeiführen eines Denial-of-Service-Zustands erheblich erschwert und andererseits eine akzeptable Auswirkung auf die Benutzererfahrung darstellt, ist eine Anmeldezeit von bis zu 30 Sekunden.

A20 Die Anzahl der gleichzeitigen Verbindungen zur Datenbank muss begrenzt werden.

STIGs: S13

Beschreibung Durch die Begrenzung der Anzahl der gleichzeitigen Verbindungen zur Datenbank wird verhindert, dass Verbindungsspitzen den Betrieb stören und zu viele Systemressourcen verbrauchen. Überdies kann dieser Zustand nicht mehr willentlich von einem Angreifer herbeigeführt werden, um einen Denial-of-Service-Zustand zu erreichen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Anzahl der gleichzeitigen Verbindungen zur Datenbank begrenzt werden kann. Der limitierende Faktor ist die maximal zulässige Anzahl an Dateideskriptoren, abzüglich eines Puffers für zusätzliche Dienste und Wartungszugänge, sodass eine realistische Grenze in Abhängigkeit zur Systemkonfiguration zwischen 60 000 und 80 000 parallelen Verbindungen liegt.

A21 Die Anzahl der parallel aktiven Sitzungen pro Benutzer muss begrenzt werden.

STIGs: S46

Beschreibung Ähnlich wie zuvor bei der Anforderung A20 muss auch die Anzahl der parallelen Verbindungen pro Benutzerkonto begrenzt werden. Dies schützt vor einer Überlastung der Systemressourcen durch einen einzelnen Benutzer und stellt sicher, dass der Zugriff auf für den Betrieb des Datenbanksystems kritische Konten dauerhaft gewährleistet werden kann.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Anzahl der gleichzeitigen Verbindungen zur Datenbank pro Benutzer begrenzt werden kann. Es ist zu beachten, dass ein Benutzer möglicherweise mehrere Schnittstellen parallel verwenden kann, um eine Verbindung herzustellen. In der Regel sollten alle Konten des Datenbanksystems nicht mehr als 80 % der insgesamt möglichen Verbindungsvolumina in Anspruch nehmen können, sofern mehrere Datenbanken und Konten im aktiven Betrieb verwendet werden. Die Verteilung der Ressourcen auf die verschiedenen Konten kann z. B. unter Berücksichtigung der Priorisierung der Daten und/oder der für diese zu erwartenden Spitzenlast vorgenommen werden.

A22 Die Zwischenspeicherung von Authentifizierungsdaten muss deaktiviert werden.

STIGs: S37, S86

Beschreibung Die Durchführung eines Authentifizierungsvorgangs ist aufgrund der damit verbundenen kryptografischen Berechnungen ein vergleichsweise kostenintensiver Vorgang. Durch eine vorübergehende Speicherung der Authentifizierungsdaten oder auch nur der Authentifizierungsentscheidung (innerhalb eines strikten Autorisierungskontextes) kann daher die Wirtschaftlichkeit verbessert werden. Gleichzeitig stellt dies ein Sicherheitsrisiko dar, da

sich ein Angreifer z. B. bei falscher/unzureichender Konfiguration oder durch nicht geschlossene Sicherheitslücken Zugang zu diesen Daten verschaffen kann. Darüber hinaus werden zwischengespeicherte Authentifizierungsdaten häufig zur Überbrückung von Ausfallzeiten der angeschlossenen IAM-Systeme verwendet, worauf ein Angreifer durch einen Denial-of-Service-Angriff abzielen kann, um die Authentifizierung zu umgehen oder die Verwendung veralteter Authentifizierungsinformationen zu erzwingen. Aus Sicherheitsgründen sollte daher die Zwischenspeicherung von Authentifizierungsdaten deaktiviert und ein Sitzungs-Timeout für zumindest alle administrativen Zugänge konfiguriert werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob ein Sitzungs-Timeout konfiguriert werden kann. Ein guter Wert, der einerseits die Sicherheit in ausreichendem Maße berücksichtigt und andererseits eine akzeptable Auswirkung auf die Benutzererfahrung darstellt, ist eine Sitzungszeit von bis zu 5 Minuten. Wird die Authentifizierung mittels Simple Authentication and Security Layer (SASL) und LDAP an einem IAM-System vorgenommen, so muss darüber hinaus die Zwischenspeicherungszeit des SASL-Servers entsprechend angepasst bzw. die Zwischenspeicherung deaktiviert werden.

3.2.4 Autorisierung

A23 Die Autorisierung muss konfiguriert und aktiv sein.

CIS: C5 | **STIGs:** S10, S48

Beschreibung Neben der Authentifizierung muss auch die Autorisierung aktiviert und konfiguriert werden. Während die Authentifizierung dazu dient, die Identität des Benutzers oder IT-Systems festzustellen, bestimmt die Autorisierung, auf welche Ressourcen die Entität zugreifen darf. Eine gewisse Form der Autorisierung muss daher in jedem Fall implementiert werden, zumindest um administrative oder sicherheitsrelevante Berechtigungen von denen für den normalen Betrieb zu trennen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine Form der Autorisierung aktiviert und konfiguriert werden kann.

A24 Jeder Benutzer muss einer Berechtigungsgruppe/Access Control List zugewiesen sein.

STIGs: S14, S47, S49

Beschreibung Durch die Verwendung von Access Control List (ACL) wird eine bestimmte Art der Autorisierung umgesetzt. Es wird festgelegt, auf welche Rechte oder Systemressourcen durch die jeweilige Zugriffskontrollliste zugegriffen werden darf. Eine Unterteilung anhand von drei ACL kann z. B. in grundlegende Systemrechte, Zugriff auf Protokolldaten und Zugriff auf administrative Funktionen erfolgen. Durch die Zuordnung von Konten zu einer oder mehreren ACL kann so eine feingranulare und gleichzeitig einheitliche Zuweisung von Zugriffsrechten vorgenommen werden. Indem die Rechtevergabe ausschließlich auf diese Weise und damit nicht über eine direkte Zuordnung zu einem Konto erfolgt, wird der Prozess transparent und übersichtlich gestaltet. Bei richtiger Konfiguration erhöht sich dadurch die Gesamtsicherheit des Systems, da die Vergabe von undokumentierten Individualberechtigungen vermieden wird.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Verwendung von ACL aktiviert und konfiguriert werden kann. Jedes Benutzerkonto muss mindestens einer ACL zugeordnet sein und darf keine Individualberechtigungen aufweisen.

A25 Benutzerkonten, die über einen längeren Zeitraum inaktiv sind, müssen deaktiviert werden.

BSI: B1

Beschreibung Inaktive Benutzerkonten können ein erhebliches Sicherheitsrisiko darstellen, da diese weiterhin über gültige Berechtigungen verfügen und aufgrund ihrer verwaisten Existenz häufig an Sicherheitsprozessen vorbei weiterbestehen. Ein Angreifer oder sogar ein ehemaliger Mitarbeiter (abhängig vom möglichen Zugang zum System) kann diese Konten nutzen, um verschiedene Arten von Angriffen durchzuführen. Aus diesem Grund müssen inaktive Benutzerkonten zeitnah deaktiviert werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Benutzerkonten automatisch nach einer vordefinierten Zeitspanne, ausgehend von der letzten Verwendung/Anmeldung, deaktiviert werden können. Dabei ist zu beach-

ten, dass Konten, die für die Verwaltung oder den fortlaufenden Betrieb der angeschlossenen Anwendungen verwendet werden, von dieser Maßnahme ausgeschlossen sind. Ein guter Wert, der einerseits die Sicherheit in ausreichendem Maße berücksichtigt und andererseits eine akzeptable Auswirkung auf die Benutzererfahrung darstellt, ist ein zulässiger Inaktivitätszeitraum von bis zu 90 Tagen. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A26 Benutzerkonten und -gruppen, die deaktiviert sind/nicht verwendet werden, müssen gelöscht werden.

BSI: B3 | **CIS:** C8

Beschreibung Als Fortsetzung der vorherigen Anforderung A25 müssen sowohl Benutzerkonten als auch -gruppen, die nicht mehr benötigt werden, gelöscht werden. Auf diese Weise wird sichergestellt, dass keine veralteten Berechtigungsgruppen versehentlich weiter verwendet werden können. Darüber hinaus werden Speicherressourcen freigegeben und unnötige Verkettungen/-Abhängigkeiten in der Berechtigungslandschaft beseitigt.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob deaktivierte Benutzerkonten und verwaiste -gruppen automatisch nach einer vordefinierten Zeitspanne gelöscht werden können. Ein guter Wert, der einerseits die Sicherheit in ausreichendem Maße berücksichtigt und andererseits eine akzeptable Auswirkung auf die Benutzererfahrung darstellt, ist ein zulässige Speicherdauer von bis zu 365 Tagen. Bezogen auf die Benutzerkonten wird diese Sicherheitsfunktionalität jedoch zumeist nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A27 Die vordefinierten Benutzerrollen sind so weit wie möglich zu verwenden.

CIS: C75

Beschreibung Die meisten Datenbanksysteme verfügen über vordefinierte Benutzerrollen. Falls vorhanden, sollten diese möglichst im operativen Einsatz verwendet werden, da diese in der Regel bereits für eine sichere Bewältigung gängiger Anwendungsszenarien konzipiert und getestet wurden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob vordefinierte Benutzerrollen zur Verfügung stehen und ob diese verwendet werden können.

A28 Die vordefinierten Benutzerrollen sind auf ihre Vereinbarkeit im Hinblick auf alle Anforderungen zu prüfen.

CIS: C11, C12, C24

Beschreibung Als Fortsetzung der vorherigen Anforderung A27 müssen die vordefinierten Benutzerrollen, sofern vorhanden, auf ihre Kompatibilität mit allen Anforderungen dieses Vorgehensmodells überprüft und gegebenenfalls angepasst werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die vordefinierten Benutzerrollen kompatibel zu allen Anforderungen des Vorgehensmodells sind. Müssen diese angepasst werden, ist besonders darauf zu achten, dass der Zugang zu den Verwaltungsfunktionen nicht durch die versehentliche Missachtung der notwendigen Reihenfolge der Berechtigungsumstrukturierung für alle Benutzer gesperrt wird.

A29 Die gleiche Benutzerkennung darf nicht von mehreren Personen oder Diensten verwendet werden.

BSI: B31

Beschreibung Um die Zurückverfolgbarkeit jeder am Datenbanksystem ausgeführten Aktivität zu einem Benutzerkonto zu gewährleisten, darf dieselbe Benutzerkennung niemals von mehr als einer Person oder einem Dienst verwendet werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Verwendung von Benutzerkonten auf eine Entität beschränkt werden kann. Im Zusammenhang mit Konten für den Zugriff durch Dienste oder der angeschlossenen Anwendung kann der Zugriff bei vielen Datenbanksystemen oft durch eine Limitierung auf der Grundlage der zugreifenden IP-Adresse beschränkt werden. Die Durchsetzung der Maßnahme im Rahmen von Personenkonten kann dagegen zumeist nur durch organisatorische, prozessuale Maßnahmen realisiert werden, da hierbei in der Regel eine dynamische Vergabe von IP-Adressen erfolgt.

A30 Die gleiche Benutzerkennung darf nicht für den Zugriff auf mehrere Datenbanken verwendet werden.

STIGs: S18

Beschreibung Für das Datenbanksystem muss eine strikte Trennung der Berechtigungen vorgenommen werden. Neben der Trennung der administrativen Berechtigungen von denen für den normalen Betrieb muss für den Zugriff auf jede Datenbank ein separates Konto angelegt werden. Dadurch wird es einem Angreifer erschwert, im Falle einer Kompromittierung auf weitere Daten unbefugt zuzugreifen, was die möglichen Auswirkungen eines Angriffs verringert.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die gleiche Benutzerkennung für den Zugriff auf mehrere Datenbanken verwendet wird. Dazu müssen die Zuordnungen der Benutzerkonten zu den einzelnen Datenbanken auf Überschneidungen überprüft werden.

A31 Eine rollenbasierte Zugriffskontrolle zur Trennung von Benutzer- und Datenbankverwaltungsfunktionen muss umgesetzt werden.

BSI: B2, B6, B71, B72 **CIS:** C21 | **STIGs:** S96, S15, S72, S19, S75, S34, S70, S50, S51

Beschreibung Die Benutzergruppen bzw. die ACL müssen mindestens so konfiguriert werden, dass eine strikte Trennung der administrativen Funktionen von denen für den normalen Betrieb/Zugriff auf die einzelnen Datenbanken vorgenommen wird. Darüber hinaus ist es in der Regel sinnvoll, eine weitere Unterteilung der Berechtigungen vorzunehmen, um z. B. einem Team von Administratoren nur den Zugriff auf die jeweils benötigten administrativen Funktionen zu gewähren. Auf diese Weise kann beispielsweise auch die Integrität von Audit-Informationen gewahrt werden, indem ein Administrator standardmäßig nicht dazu berechtigt wird, diese einzusehen und zu bearbeiten.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die explizit den Datenbanken zugewiesenen Benutzerkonten über administrative Rechte verfügen oder Benutzergruppen/ACL zugeordnet sind, die solche Rechte verleihen. Ist dies der Fall, muss das bestehende Berechtigungskonzept entsprechend angepasst werden.

A32 Die Vergabe von Zugriffsrechten muss nach dem Least-Privilege- und Erforderlichkeitsprinzip erfolgen.

BSI: B4 | **CIS:** C20, C23, C70, C71, C72

Beschreibung Als Fortsetzung bzw. Konkretisierung der vorangegangenen Anforderungen A24, A27, A28, A30 und A31 müssen alle Benutzerrollen/ACL auf ihre Konformität mit dem Grundsatz der geringstmöglichen Berechtigungen und der Erforderlichkeit hin überprüft werden. Der Grundsatz der geringstmöglichen Berechtigungen (englisch „Principle of Least Privilege (PoLP)“) besagt, dass die Zugriffsrechte eines Benutzers oder einer Anwendung auf das erforderliche Minimum für die Bewältigung der vorgesehenen Aufgaben beschränkt werden müssen. Das Erforderlichkeitsprinzip (englisch „Need-to-know“) besagt, dass einem Benutzer oder einer Anwendung kein Zugang zu Informationen oder Daten gewährt werden darf, wenn diese nicht unmittelbar für die Bewältigung der vorgesehenen Aufgaben erforderlich sind. Die Berücksichtigung beider Prinzipien kann die Auswirkungen einer Kompromittierung des Datenbanksystems erheblich verringern, indem die Möglichkeit der Ausweitung von Privilegien eingeschränkt und der Umfang eines möglichen Datenzugriffs reduziert wird.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die vorhandenen Benutzergruppen/ACL dem Least-Privilege- und Erforderlichkeitsprinzip entsprechen.

3.2.5 Passwortrichtlinien

A33 Verwendete Passwörter müssen hohen Sicherheitsanforderungen standhalten.

BSI: B9, B12, B13 | **STIGs:** S115

Beschreibung Passwörter sind einer Vielzahl möglicher Angriffe ausgesetzt, darunter Brute-Force- oder Wörterbuchangriffe. Unter Berücksichtigung der maximalen Länge der bereits verfügbaren Rainbow-Tabellen, der Geschwindigkeit modernster Hardware für kryptografische Berechnungen und einer ausgewogenen Berücksichtigung der Benutzererfahrung müssen Passwörter die folgenden Anforderungen erfüllen, um langfristig als sicher zu gelten:

1. Eine Mindestlänge von 14 oder mehr Zeichen.
2. Das Enthalten von Zeichen aus mindestens drei der folgenden Kategorien:
   - Ziffern (0 – 9)
   - Großbuchstaben (A – Z)
   - Kleinbuchstaben (a – z)
   - Regionale Sonderzeichen (z. B. ä, ç, ö, ß, ü) oder Unicode-Zeichen
   - Sonderzeichen (z. B. !, $, %, @, ␣)
3. Keine Merkmale des Benutzernamens oder der Entität, die mit dem Benutzerkonto verbunden ist (z. B. Vorname, Geburtsdatum bzw. Servername, Fully Qualified Domain Name (FQDN)).
4. Keine Bestandteile früherer Passwörter enthalten (optional, da eine sichere Umsetzung nicht in jedem Fall gewährleistet werden kann).

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine Kennwortrichtlinie konfiguriert und durchgesetzt werden kann. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A34 Vordefinierte Standard-Passwörter und -Benutzerkennungen dürfen nicht verwendet werden.

BSI: B11, B16, B44 | **CIS:** C7

Beschreibung Viele Anwendungen, darunter auch Datenbanksysteme, werden mit einem vorkonfigurierten Standard-Benutzer und -Passwort ausgeliefert bzw. automatisch eingerichtet. Diese sind in der Regel dazu gedacht, einen schnellen Einstieg in die Konfiguration und Nutzung der Anwendung zu ermöglichen, indem eine funktionale minimale Erstkonfiguration ohne Benutzereingriff automatisiert abgeschlossen werden kann. Dieser Umstand birgt jedoch ein hohes Sicherheitsrisiko, da die Standard-Zugangsdaten über alle Installationen der Anwendung hinweg identisch sowie öffentlich dokumentiert sind und solche Anwendungen zudem leicht identifiziert werden können. Wenn ein Angreifer in der Lage ist, auf das System zuzugreifen, kann er sich problemlos authentifizieren und eine weitreichende Kompromittierung herbeiführen, da

der Standard-Benutzer in der Regel immer auch über administrative Rechte verfügt.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob vordefinierte Standard-Passwörter und -Benutzerkennungen vorhanden sind. Ist dies der Fall, müssen alle entsprechenden Benutzerkonten deaktiviert, gelöscht oder umbenannt und die zugehörigen Standard-Passwörter entsprechend der Anforderung A33 geändert werden.

A35 Passwörter und kryptografische Schlüssel dürfen nur einen einzigen Einsatzzweck aufweisen und nicht mehrfach verwendet werden.

BSI: B8, B37, B38

Beschreibung Die Verwendung identischer Passwörter oder kryptografischer Schlüssel für mehrere Benutzerkonten kann im Rahmen von Brute-Force-Angriffen ein hohes Sicherheitsrisiko darstellen. Wenn ein Angreifer gültige Anmeldedaten für ein Benutzerkonto ermittelt hat, wird mit hoher Wahrscheinlichkeit bei gleichartigen Konten die Verwendung des gleichen Kennworts erprobt. Ein Beispiel für gleichartige Konten sind das Standard- und das Administrator-Konto eines Mitarbeiters, z. B. wenn diese anhand ihrer Namensgebung miteinander in Verbindung gebracht werden können. Aus diesem Grund muss die Verwendung identischer Anmeldedaten für mehrere Benutzerkonten unterbunden werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob für mehrere Benutzerkonten das identische Passwort oder ein identischer kryptografischer Schlüssel verwendet wird. In den meisten Fällen kann diese Sicherheitsfunktionalität jedoch nicht direkt über das DBMS verifiziert werden, da zumindest in Bezug auf Passwörter diese durch ein angeschlossenes IAM-System verwaltet werden. Werden diese jedoch lokal gespeichert, kann die Prüfsumme aller Passwörter unter Berücksichtigung des Passwort-Salts bzw. -Peppers verglichen bzw. kann das referenzierte Zertifikat auf eine Mehrfachverwendung untersucht werden.

A36 Frühere Passwörter dürfen nicht wiederverwendet werden.

BSI: B17

Beschreibung Die Wiederverwendung von Passwörtern mindert die Wirksam-

keit von Maßnahmen zur Passwortsicherheit erheblich. Bestimmte Anwender neigen dazu, Passwörter aus Komfortgründen/Gewohnheit wiederzuverwenden oder im Rahmen einer Passwortänderung nur Teile auszutauschen, z. B. das erste oder letzte Zeichen. Dies ermöglicht sogenannte Credential-Stuffing-Angriffe, bei denen bekannte Zugangsdaten aus früheren erfolgreichen Angriffen oder Datenverletzungen durch einen Angreifer wiederverwendet werden. Diese Art von Brute-Force-Angriff ist heute eine der am häufigsten verwendeten Techniken zur Übernahme von Benutzerkonten, was mitunter auf die hohe Erfolgsquote zurückzuführen ist. Daher muss die Wiederverwendung von Passwörtern wirksam unterbunden werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Wiederverwendung von Passwörter unterbunden werden kann. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A37 Passwörter dürfen nur mit einer sicheren Methode als Hash unter Verwendung eines Salts sowie, falls möglich, mit Peppering gespeichert werden.

BSI: B18, B24, B27, B51 | **STIGs:** S22, S85

Beschreibung Passwörter dürfen niemals im Klartext gespeichert werden, sondern müssen durch kryptografische Hash-Funktionen vor unberechtigtem Zugriff geschützt werden. Im Falle einer Kompromittierung des Datenbanksystems bleibt einem Angreifer somit nur die Möglichkeit, die zugehörigen Eingabewerte durch einen Brute-Force-Angriff zu ermitteln. Aufgrund der Tatsache, dass in diesem Angriffskontext ein unmittelbarer Zugriff auf die Hash-Werte vorhanden ist, wodurch wesentlich leistungsstärkere Angriffsmethoden zur Verfügung stehen, muss bei der Auswahl der Hash-Funktionen darauf geachtet werden, besonders sichere und damit ressourcenintensive Verfahren auszuwählen. In BSI-TR-02102-1 werden die folgenden Hash-Funktionen empfohlen:

- SHA-256, SHA-512/256, SHA-384 und SHA-512 (siehe auch [84]),
- SHA3-256, SHA3-384, SHA3-512 (siehe auch [85]).

Als weitere Schutzmaßnahme muss zur Berechnung des Hash-Werts jedem

Passwort ein individueller Salt (zufällig generierte Zeichenfolge) beigefügt werden. Ein Angreifer ist im Zuge eines Brute-Force-Angriffs dadurch gezwungen, alle Passwortkombinationen für jeden entwendeten Passwort-Hash einzeln zu berechnen. Dies erhöht den erforderlichen Rechenaufwand proportional zur Anzahl der zu brechenden Hash-Werte und unterbindet die Verwendung einer Rainbow-Tabelle. Darüber hinaus lässt sich durch einen Vergleich aller Hash-Werte nicht länger nachvollziehen, ob mehrere Benutzer ein identisches Passwort verwenden.

Als ergänzende Schutzmaßnahme kann neben einem individuellen Salt auch Peppering eingesetzt werden. Im Unterschied zum Salt wird dabei ein statischer Schlüssel verwendet, der ebenfalls in die Berechnung der Hash-Werte einbezogen wird, jedoch außerhalb der Datenbank gespeichert ist, z. B. in einer geschützten Konfigurationsdatei oder in einem Hardware-Sicherheitsmodul. Wird die Datenbank kompromittiert, kann ein Angreifer somit keine Passwortangriffe durchführen, ohne seinen Zugriff über das Datenbanksystem hinaus auszudehnen, was im Falle einer vollständigen Systemhärtung eine hohe zusätzliche Hürde darstellt. In BSI-TR-02102-1 wird das HMAC-Verfahren in Kombination mit den genannten Hash-Funktionen für Peppering empfohlen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die aufgeführten Hash-Funktionen unter Verwendung von Salts und ggf. Peppering für die lokale Speicherung der Zugangsdaten zum Datenbanksystem unterstützt werden und konfiguriert werden können. Darüber hinaus muss geprüft werden, ob die bereits vorhandenen Zugangsdaten nach der Konfigurationsanpassung automatisch an die neuen Anforderungen angepasst werden.

A38 Mechanismen zum Zurücksetzen von Passwörtern dürfen keine Angriffsfläche für Angreifer bieten.

BSI: B23

Beschreibung Gelegentlich stellen auch Datenbanksysteme Mechanismen zur Verfügung, die den Zugriff auf Verwaltungsfunktionen ohne die Kenntnis von Zugangsdaten ermöglichen. Diese werden in der Regel verwendet, um im Falle von Authentifizierungsproblemen oder anderen Störungen den Zugang wiederherstellen zu können. Daher muss sichergestellt werden, dass diese nicht von einem Angreifer ausgenutzt bzw. unberechtigt aktiviert werden können.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Mechanismen

zum Zurücksetzen von Passwörtern vorhanden sind und wie diese speziell vor unberechtigtem Zugriff geschützt werden können. Dies gilt insbesondere dann, wenn diese nicht ausschließlich durch ein angeschlossenes IAM-System bereitgestellt werden, sondern lokal direkt auf das Datenbanksystem angewendet werden können.

A39 Passwörter dürfen nicht aufgrund von zeitlichen Nutzungsbegrenzungen geändert werden.

BSI: B14

Beschreibung Das regelmäßige Ablaufen von Passwörtern nach einem vordefinierten Zeitraum, d. h. ohne weiteren Grund, ist eine mittlerweile überholte Praxis, da sich die zugrunde liegende Bedrohungssituation signifikant verändert hat. Das größte Risiko für Passwörter besteht nicht mehr in der Berechnung des zugehörigen Passwort-Hashes, insbesondere wenn die Anforderung A37 korrekt umgesetzt wurde. Stattdessen werden Anmeldedaten durch gezieltes Phishing, Social-Engineering-Angriffe oder auch Spionagesoftware entwendet und innerhalb weniger Stunden zur Durchführung von Angriffen mit hohem Automatisierungsgrad missbraucht. Die Erzwingung regelmäßiger Passwortänderungen hat daher heute nur noch einen psychologischen Effekt, stellt aber in der Praxis keinen Sicherheitsgewinn dar. Vielmehr ist dies sogar mit negativen Effekten verbunden:

1. Anwender werden dazu verleitet, nur Teile des Passworts auszutauschen, z. B. das erste oder letzte Zeichen.
2. Anwender werden zum Aufschreiben von Passwörtern bewegt, wodurch diese weniger geschützt sind.
3. Es entstehen Kosten in Form von verlorener Arbeitszeit und Anfragen beim Service Desk.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Zugangsdaten automatisch nach einer vordefinierten Zeitspanne auslaufen. Ist dies der Fall, so muss diese Funktion deaktiviert werden. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch ein angeschlossenes IAM-System entsprechend umgesetzt werden.

A40 Zur Erkennung von Passwortkompromittierungen müssen geeignete Schutzmaßnahmen ergriffen werden.

BSI: B15

Beschreibung Zur Aufdeckung von Passwortkompromittierungen können Präventivmaßnahmen ergriffen werden. Dazu gehört z. B. die Protokollierung aller Interaktionen mit dem Datenbanksystem (siehe Anforderungen A41 und A47). Darüber hinaus können aber auch weitergehende Maßnahmen ergriffen werden, wie z. B. der automatisierte Abgleich von Zugangsdaten mit der „Have I Been Pwned?“ Datenbank oder die Anzeige von sicherheitsrelevanten Informationen für den Benutzer bei der Nutzung von Management-Schnittstellen (Zeitpunkt und Quell-IP-Adresse des letzten Zugriffs).

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Maßnahmen zur Detektion von Passwortkompromittierungen umgesetzt werden können. Viele Maßnahmen werden zumeist jedoch nicht direkt vom DBMS bereitgestellt, sondern müssen durch ein angeschlossenes IAM-, Intrusion Detection System (IDS)/Intrusion Prevention System (IPS)- oder auch Security Information and Event Management (SIEM)-System entsprechend umgesetzt werden.

3.2.6 Auditierung und Protokollierung

A41 Die Auditierung sowie Protokollierung muss konfiguriert und aktiv sein.

BSI: B62 | **CIS:** C13, C14, C31, C68 | **STIGs:** S1, S52, S56, S114

Beschreibung Um Ereignisse und Aktivitäten bei der Nutzung oder Verwaltung des Datenbanksystems zu verfolgen, müssen Sicherheitsprotokolle erstellt und gespeichert werden. Dabei handelt es sich um detaillierte, textbasierte Aufzeichnungen, die zur Identifizierung verdächtiger oder irregulärer Aktivitäten bzw. zur Aufdeckung von Verstößen gegen (interne) Richtlinien verwendet werden können. Sie bilden die Grundlage für Sicherheitsanalysen durch Blue-Teams oder Forensiker und sind daher ein äußerst wichtiger Bestandteil einer jeden Absicherungsstrategie.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Protokol-

lierung von Ereignissen im Datenbanksystem erfolgen und aktiviert werden kann.

A42 Sind mehrere Audit- und Protokollierungsfunktionen verfügbar, ist das sicherste Verfahren zu verwenden.

CIS: C43, C51, C52, C53, C54, C55

Beschreibung Je nach Datenbanksystem und dessen Reifegrad können mehrere Methoden zur Protokollierung von Ereignissen und Aktivitäten zur Verfügung stehen. Im Wesentlichen wird zwischen der Protokollierung in eine entsprechende Protokolldatei und der Verwendung des Syslog-Protokolls bzw. systemd/Journal unterschieden. Die Verwendung von Syslog oder systemd/-Journal hat im Kontext des beschriebenen Vorgehensmodells im Wesentlichen die folgenden Vorteile:

- Die Standard-Konfiguration schützt die Protokolldaten bereits sowohl vor unbefugtem Zugriff als auch vor Manipulationen.
- Die Anbindung an ein externes Log-Management- oder SIEM-System wird vereinfacht.
- In Abhängigkeit von der Umsetzung im Datenbanksystem kann die Protokollierung leicht individuell angepasst werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, welche Audit- und Protokollierungsfunktionen zur Verfügung stehen und wie diese vor dem Hintergrund des Absicherungsziels dieses Vorgehensmodells zu bewerten sind. Grundsätzlich ist die Protokollierung per Syslog oder systemd/Journal zu bevorzugen, der Schutz vor unbefugten Zugriffen und Manipulationen kann aber auch bei Verwendung von Protokolldateien gewährleistet werden.

A43 Für Audit-Protokolle muss ausreichend Speicherplatz bereitgestellt werden.

STIGs: S32, S57

Beschreibung Um eine lückenlose Protokollierung zu gewährleisten, muss dem Datenbanksystem ausreichend Speicherplatz zugewiesen werden, der explizit für die Speicherung der Protokolldateien reserviert ist. Dies ist auch dann erforderlich, wenn, wie in Anforderung A44 empfohlen, die Protokolldateien in

ein angeschlossenes Log-Management- oder SIEM-System ausgelagert werden. Dadurch wird es möglich, Wartungsintervalle oder Störungen zu überbrücken, ohne dass es zu einem Informationsverlust bzw. zu Lücken in der Protokollierung kommt.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob genügend explizit reservierter Speicherplatz für die Speicherung von Protokolldateien vorhanden ist. Standardmäßig werden Protokolldateien unter /var/log abgelegt, sowohl bei der Verwendung einer eigenständigen Protokolldatei als auch bei der Verwendung von Syslog bzw. systemd/Journal. Der Bedarf an Speicherplatz muss individuell auf der Grundlage der Gesamtanzahl der Benutzer, der Nutzungsintensität, der eingestellten Log-Verbosität (siehe Anforderung A47) und des eingestellten Auslagerungsintervalls bei der Verwendung eines externen Log-Management- oder SIEM-Systems bestimmt werden. Angesichts der Tatsache, dass in Abhängigkeit von diesen Faktoren große Datenmengen erreicht werden können, einige Anforderungen bei fehlerhafter Protokollierung nicht mehr erfüllt werden und Speicherplatz als Kostenfaktor an dieser Stelle eine zu vernachlässigende Rolle spielt, sollte der reservierte Speicherplatz eher großzügig bemessen werden, z. B. mit 50 – 100 GB.

A44 Audit-Protokolle müssen in ein separates Log-Management-System ausgelagert werden.

CIS: C44 | **STIGs:** S31, S55, S58

Beschreibung Die zyklische regelmäßige Auslagerung von Protokolldaten an ein externes Log-Management- oder SIEM-System ist mit mehreren Vorteilen verbunden. Zum einen wird lokaler Speicherplatz freigesetzt, sodass die Erfüllung der Anforderung A43 auch langfristig gewährleistet werden kann. Zum anderen werden die vorhandenen Protokolldaten effektiver vor versehentlicher Änderung oder Löschung geschützt. Im Kontext eines SIEM-Systems führt die dortige Normalisierung und Korrelation erst zu einer effektiven Identifizierung von Problemen und Sicherheitsverstößen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Methoden zur Anbindung eines externen Log-Management- oder SIEM-Systems vorhanden sind. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch die lokale serverseitige Konfiguration von Logrotate implementiert werden.

A45 Um den Verlust von Audit-Protokollen zu verhindern, müssen Warnungen gesendet werden, wenn der Speicherplatz knapp wird oder die Protokollierung fehlschlägt.

STIGs: S33, S59, S60, S61

Beschreibung Durch die Konfiguration von Trigger-Warnungen können Administratoren benachrichtigt werden, wenn sich der für die Protokolldateien reservierte Speicherbereich allmählich der Kapazitätsgrenze nähert. Im Allgemeinen ist die Wahrscheinlichkeit, dass der Trigger ausgelöst wird, eher gering, kann aber aufgrund von Programmierfehlern oder Fehlern bei der Verarbeitung durch Syslog bzw. systemd/Journal oder auch Logrotate auftreten.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Konfiguration von Trigger-Warnungen für den Fall des Überschreitens von Kapazitätsgrenzen vorgenommen werden kann. Ein guter Wert, ab dem eine Trigger-Warnung ausgegeben werden kann, ist das Überschreiten von 75 % der verfügbaren Speicherplatzkapazität für Protokolldateien.

A46 Audit-Protokolle sind geordnet nach ihrem Alter zu überschreiben, wenn der Speicherplatz für neue Einträge erschöpft ist.

CIS: C33, C48, C49, C50 | **STIGs:** S62

Beschreibung Für den Fall, dass die vorangegangenen Maßnahmen zur Vermeidung der vollen Speichernutzung aus unerwarteten Gründen fehlschlagen, müssen Maßnahmen für die daraus resultierenden Protokollierungsfehler definiert werden. Bei lokaler Protokollierung kann das First In –- First Out (FIFO)-Prinzip befolgt werden, wonach die ältesten Protokollierungsdaten zuerst überschrieben werden. Dies ist allerdings mit einem Informationsverlust verbunden und sollte vermieden werden. Werden Protokolldaten ausgelagert, sollten diese so weit wie möglich erhalten bleiben, indem z. B. der Logrotate-Dienst so konfiguriert wird, dass er im Falle eines Fehlers automatisch neu startet oder im Falle von Verbindungsfehlern automatisch eine neue Verbindung zum Log-Management- oder SIEM-System aufbaut und ausstehende Synchronisierungen kompensiert.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob das FIFO-Prinzip oder andere Kompensationsmaßnahmen für den Fall einer vollen Spei-

cherauslastung umgesetzt werden können. In den meisten Fällen wird diese Sicherheitsfunktionalität jedoch nicht direkt vom DBMS bereitgestellt, sondern muss durch die lokale serverseitige Konfiguration von Syslog bzw. systemd/-Journal oder auch Logrotate implementiert werden.

A47 Audit-Protokolle müssen alle Ereignisse und Aktivitäten erfassen (maximale Verbosität).

BSI: B61, B67 | **CIS:** C32, C61, C62, C63, C64, C66, C56, C82 | **STIGs:** S54, S102

Beschreibung Um die Auswertung sicherheitsrelevanter Ereignisse möglichst effektiv und umfassend nachvollziehen zu können, müssen Protokollierungen so ausführlich wie möglich erzeugt und abgespeichert werden. Das Zurückhalten von Log-Informationen dient in der Regel dazu, den notwendigen Speicherplatz zu reduzieren, steht aber in keinem Verhältnis zu der damit verbundenen Behinderung des Analyseprozesses. Aus diesem Grund sollte die Protokollierungsstufe (englisch „Log level“) so hoch wie möglich eingestellt werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Protokollierungsstufe angepasst und somit angehoben werden kann, um so viele Informationen über Ereignisse und Aktivitäten wie möglich vorhalten zu können.

A48 Audit-Protokolle müssen einem vordefinierten Format entsprechen, das ihre Analyse erleichtert.

BSI: B52 | **CIS:** C65, C67 | **STIGs:** S63

Beschreibung Das für die Protokollierung verwendete Format muss einem anerkannten Standard entsprechen, damit eine Normalisierung und Korrelation so einfach wie möglich durchgeführt werden kann. Im Allgemeinen kann dieser Prozess immer bewältigt werden, aber im Falle größerer Abweichungen ist unter Umständen die manuelle Definition oder Implementierung eines Log-Parsers nur für das entsprechende Datenbanksystem notwendig, um die sicherheitsrelevanten Informationen zu extrahieren. Dies sollte aufgrund des damit verbundenen Aufwands vermieden werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob ein anerkannter Standard für die Erzeugung von Protokollinformationen verwendet bzw. konfiguriert werden kann. Die Verwendung von Syslog bzw. systemd/Journal

erfüllt bereits einen großen Teil der Anforderungen, da der in RFC 3164 standardisierte Header automatisch verwendet wird, welcher weit verbreitet ist. Die verbleibenden Anforderungen an die Attribute des Protokolleintrags werden dann in der Regel auch vom Hersteller korrekt umgesetzt, da hierfür die zugrunde liegende Programmierschnittstelle herangezogen wird.

A49 Audit-Protokolle müssen in einem Verzeichnis mit leicht zuzuordnenden Dateinamen gespeichert werden.

CIS: C45, C46

Beschreibung Standardmäßig werden die Protokolldateien unter /var/log gespeichert. Vorzugsweise sollte für die Protokolle des Datenbanksystems ein eigener Unterordner angelegt werden, da dies das Auffinden der zugehörigen Protokolldateien und die Konfiguration von Logrotate vereinfacht. Außerdem sollte die Bezeichnung der Protokolldateien ein Präfix mit den folgenden Informationen enthalten:

- Der Zeitstempel, zu dem die Protokolldatei erstellt wurde (entspricht dem Zeitpunkt der frühesten enthaltenen Protokollinformation).
- Der Name des Datenbanksystems oder eine entsprechende Abkürzung.
- Der Name der Datenbanksystemkomponente oder eine entsprechende Abkürzung, falls je nach Funktion mehrere Protokolldateien erstellt werden.
- Falls erforderlich, eine fortlaufende Nummer.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob der Speicherort und die Bezeichnung der Protokolldateien angepasst werden kann.

A50 Der Zugriff auf die Konfiguration der Auditierung und Protokollierung muss begrenzt werden.

STIGs: S53

Beschreibung Wird der Zugriff auf die Konfiguration der Auditierung und Protokollierung nicht eingeschränkt, können unbefugte Entitäten die Erfassung von Ereignissen und Aktivitäten verhindern oder beeinträchtigen. Auf diese Weise kann ein Angreifer die Aufdeckung seiner Angriffe umgehen, da wichtige Schlüsselinformationen zur Schaffung eines vollständigen Lagebilds verloren gehen, die zur Durchführung einer Korrelation notwendig sind. Dar-

über hinaus kann die Entstehung eines Denial-of-Service-Zustands durch eine vorsätzliche Fehlkonfiguration provoziert werden, z. B. durch die Aktivierung der Aufzeichnung aller Debug-Informationen zusätzlich zu einer hohen Verbosität (vgl. Anforderung A47), sodass signifikant mehr Speicherplatz beansprucht wird als im Systemdesign vorgesehen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob der Zugriff auf die Konfiguration der Auditierung und Protokollierung durch eine Anpassung der bestehenden Benutzergruppen/ACL eingeschränkt werden kann bzw. ob die bestehende Berechtigungsvergabe dem Least-Privilege- und Erforderlichkeitsprinzip aus Anforderung A32 entspricht.

A51 Der Zugriff auf die Inhalte der Auditierung und Protokollierung muss begrenzt werden.

STIGs: S29

Beschreibung Neben der Beschränkung des Zugriffs auf die Konfiguration der Auditierung und Protokollierung muss außerdem gewährleistet sein, dass auch die Protokollierungsinhalte nur von einer begrenzten Gruppe von Berechtigungen eingesehen werden können. Der Zugriff auf detaillierte Protokollierungsdaten kann von einem Angreifer eingesetzt werden, um interne Anwendungsinformationen abzugreifen oder auch die Wirksamkeit von Angriffen zu verifizieren/erhöhen. Darüber hinaus kann der Umfang der Protokollierung als Grundlage für die Festlegung der maximalen Angriffsintensität in Relation zur Entdeckungswahrscheinlichkeit dienen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob der Zugriff auf die Protokollierungsinhalte durch eine Anpassung der bestehenden Benutzergruppen/ACL eingeschränkt werden kann bzw. ob die bestehende Berechtigungsvergabe dem Least-Privilege- und Erforderlichkeitsprinzip aus Anforderung A32 entspricht.

3.2.7 Monitoring

A52 Alle kritischen Parameter, Ereignisse und Betriebszustände müssen überwacht werden.

BSI: B74, B75 | **CIS:** C30

Beschreibung Um einen reibungslosen Betrieb zu gewährleisten, sollten alle kritischen Betriebszustände anhand vordefinierter Schwellenwerte überwacht werden. Abweichungen und Fehler müssen protokolliert werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine Überwachung der Betriebszustände, z. B. mittels Simple Network Management Protocol (SNMP), konfiguriert werden kann. Darüber hinaus muss der Linux-Audit-Daemon als Basis für die Protokollierung sicherheitsrelevanter Informationen installiert und konfiguriert werden. Dies ist auch eine Grundbedingung für die Verwendung von Syslog oder systemd/Journal (vgl. Anforderungen A42, A43 und A46).

3.2.8 Fingerprinting

A53 Der Debug-Modus muss deaktiviert werden.

CIS: C57, C58, C59, C60

Beschreibung Ein aktiver Debug-Modus im produktiven Betrieb kann je nach Datenbanksystem dazu führen, dass sensible Informationen ausgegeben oder protokolliert werden. Dies kann ein Angreifer ausnutzen, um Angriffe durchzuführen, da sensible Informationen beispielsweise auch Zugangsdaten darstellen können.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob ein Debug-Modus zur Verfügung steht und ob dieser gegebenenfalls deaktiviert ist.

A54 Die Ausgabe von Fehlermeldungen darf nicht zur Durchführung von Angriffen interpretiert werden können.

BSI: B48 | **STIGs:** S28

Beschreibung Die Ausgabe von Fehlermeldungen ist sowohl für die Fehlersuche bei Betriebs- als auch bei Anwendungslogikfehlern notwendig. Eine unsachgemäße Handhabung kann jedoch Angreifern Ansatzpunkte für die Durchführung von Angriffen liefern, etwa bei der Ausgabe von zu vielen Informationen. Aus diesem Grund sollten die Fehlerinformationen auf das Wesentliche reduziert werden oder, wenn möglich, nur ein interner Fehlercode ausgegeben werden. Detaillierte Angaben sollten nur für authentifizierte Benutzer verfüg-

bar sein.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Ausgabe von Fehlermeldungen stark begrenzt oder deaktiviert werden kann.

A55 Die Verbindung zum Datenbanksystem darf keine Rückschlüsse auf die Version zulassen.

BSI: B56

Beschreibung Beim Verbindungsaufbau zum Datenbanksystem, gleich ob zur Management-Schnittstelle oder zur API für den Anwendungszugriff, können Anhaltspunkte zurückgegeben werden, die Rückschlüsse auf die verwendete Version zulassen. Grundsätzlich kann die Version des Datenbanksystems von einem Angreifer bei direktem Zugriff immer zumindest eingegrenzt werden, allerdings sollte dieser Vorgang so aufwändig wie möglich gestaltet werden. Bei bestehenden Schwachstellen, z. B. aufgrund von nicht installierten Aktualisierungspaketen, kann ein Angreifer anhand der Versionsnummer sehr schnell Angriffsflächen und entsprechende Exploits identifizieren, um das Datenbanksystem ggf. zu kompromittieren oder anderweitig gezielt anzugreifen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Ausgabe der Versionsnummer für nicht authentifizierte Benutzer oder ganz deaktiviert werden kann.

3.2.9 Verschlüsselung

A56 Die Kommunikation über Schnittstellen muss verschlüsselt erfolgen.

BSI: B43, B20, B25, B35, B73 | **CIS:** C80, C16 | **STIGs:** S5, S40, S109

Beschreibung Durch die Verwendung von Schnittstellen, sowohl Management- als auch Anwendungsschnittstellen/API, werden Informationen außerhalb des Datenbanksystems übermittelt oder ausgetauscht. Um die Vertraulichkeit und Integrität aller übertragenen Informationen zu gewährleisten, müssen diese stets verschlüsselt übertragen werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine verschlüsselte Übertragung für alle vorhandenen Schnittstellen konfiguriert werden kann. Insbesondere bei ASCII-basierten Anwendungsschnittstellen/Protokollen muss

eine implizite Verschlüsselung sichergestellt werden (die Verwendung der Verschlüsselung wird vom Server erzwungen).

A57 Die Kommunikation aller Teilnehmer in einem Cluster muss verschlüsselt erfolgen.

CIS: C15

Beschreibung Einige nicht-relationale DBMS unterstützen die Konfiguration eines aus mehreren Servern bestehenden Clusters. Für den Betrieb ist eine Synchronisation bzw. Replikation der Daten zwischen allen Teilnehmern des Clusters notwendig. Dazu werden Schnittstellen verwendet, um neue Daten laufend an alle Teilnehmer zu übermitteln oder Änderungen an bestehenden Daten zu melden. Um die Vertraulichkeit und Integrität aller übertragenen Daten zu gewährleisten, müssen diese stets verschlüsselt übertragen werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob für den Cluster-Betrieb eine gesonderte Konfiguration erforderlich ist, um eine verschlüsselte Übertragung an allen zu diesem Zweck verwendeten Schnittstellen zu gewährleisten.

A58 Die Verschlüsselung muss mit sicheren kryptografischen Protokollen betrieben werden.

CIS: C25, C26, C27, C79

Beschreibung Ältere Versionen des TLS-Protokolls oder seines Vorgängers SSL weisen kryptografische Schwachstellen auf, die den heutigen Sicherheitsanforderungen nicht länger Rechnung tragen. In BSI-TR-02102-2 werden die folgenden kryptografischen Protokolle empfohlen:

- TLS 1.2,
- TLS 1.3.

Umsetzung Das DBMS muss daraufhin untersucht werden, dass alle als unsicher eingestuften kryptografischen Protokolle deaktiviert sind, insbesondere SSL 2.0, SSL 3.0, TLS 1.0 und TLS 1.1, und nur sichere Protokolle wie TLS 1.2 oder TLS 1.3 verwendet werden.

A59 Das für den Schlüsselaustausch verwendete Verfahren muss sicher sein.

BSI: B39 | **STIGs:** S25

Beschreibung Ein Schlüsselaustauschprotokoll dient dem Austausch eines gemeinsamen, geheimen Schlüssels zwischen mehreren Verbindungspartnern über einen unsicheren Kommunikationskanal. Zu diesem Zweck kann ein geeignetes sicheres kryptografisches Verfahren aus dem Bereich der asymmetrischen Kryptografie verwendet werden, um einen einmaligen Schlüssel im Rahmen des Verbindungsaufbaus zu vereinbaren. In BSI-TR-02102-2 werden die folgenden Schlüsselaustauschprotokolle empfohlen:

- Elliptic Curve Diffie-Hellman Ephemeral (ECDHE),
- Diffie-Hellman Ephemeral (DHE).

Alternativ kann auch ein zuvor vereinbarter statischer Schlüssel (Pre-Shared-Key) verwendet werden, z. B. im Rahmen des Cluster-Betriebs.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Verwendung von ECDHE und/oder DHE konfiguriert und alle abweichenden Schlüsselaustauschprotokolle deaktiviert werden können. Sollte es nicht möglich sein, Perfect Forward Secrecy (PFS) als Grundbedingung für die Inanspruchnahme der ephemeren Protokollvarianten zu verwenden, ist alternativ auch die Verwendung der entsprechenden Schlüsselaustauschprotokolle ohne dieses Merkmal möglich. Diese müssen jedoch spätestens nach 2026 einer Sicherheitsbewertung unterzogen werden.

A60 Die Verschlüsselung muss mit sicheren kryptografischen Algorithmen betrieben werden.

BSI: B33, B42 | **CIS:** C28, C78 | **STIGs:** S11, S90, S92, S93, S94, S95

Beschreibung Auf der Grundlage des sicheren Schlüsselaustauschs werden sämtliche Kommunikationsverbindungen mit dem symmetrischen Verfahren Advanced Encryption Standard (AES) verschlüsselt. Hierfür stehen eine ganze Reihe von Betriebsarten zur Verfügung, die die Verschlüsselung durch Padding und die Verknüpfung von Nachrichtenblöcken steuern und sich auch in ihrer Sicherheit unterscheiden. In BSI-TR-02102-2 werden die folgenden AES-Betriebsarten empfohlen:

- AES_128_GCM und AES_256_GCM,
- AES_128_CCM und AES_256_CCM.

Von einer Verwendung von AES_128_CBC oder AES_256_CBC wird angesichts der Lucky-13-Angriffsmethode (Timing basierter Seitenkanal-Angriff auf Cipher Block Chaining (CBC)) abgeraten, bis „Encrypt-then-MAC“ Algorithmen verfügbar sind.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob AES mit den empfohlenen Betriebsarten konfiguriert werden kann.

A61 Die kryptografischen Algorithmen müssen eine hohe Schlüssellänge aufweisen.

BSI: B34

Beschreibung Neben der Verwendung sicherer kryptografischer Verfahren muss deren Sicherheit schlussendlich auch durch die Verwendung einer hohen Schlüssellänge gewährleistet werden. In BSI-TR-02102-1 werden je nach symmetrischem oder asymmetrischem Verfahren die folgenden Schlüssellängen empfohlen:

- AES und Message Authentication Code (MAC): $\geq$ 128 Bits
- Rivest-Shamir-Adleman (RSA) und DHE: $\geq$ 3000 Bits
- ECDHE und Elliptic Curve Digital Signature Algorithm (ECDSA): $\geq$ 250 Bits

Umsetzung Das DBMS muss daraufhin untersucht werden, ob alle symmetrischen und asymmetrischen Verfahren mit der erforderlichen Mindestschlüssellänge konfiguriert werden können.

A62 Selbstsignierte Zertifikate dürfen für eine verschlüsselte Kommunikation nicht verwendet und akzeptiert werden.

BSI: B40 | **STIGs:** S38, S87, S100

Beschreibung Für den produktiven Einsatz müssen gültige Zertifikate verwendet werden, die von einer anerkannten oder unternehmenseigenen Zertifizierungsstelle generiert und signiert wurden. Dadurch wird unter anderem die Integrität im Zuge des Authentifizierungsprozesses sichergestellt.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die verwendeten Zertifikate von einer offiziellen Zertifizierungsstelle stammen und nicht abgelaufen sind.

A63 Für die Erstellung von Anmeldeinformationen und Zertifikaten müssen sichere Schlüsselgeneratoren verwendet werden.

BSI: B36

Beschreibung Zur Erzeugung von Zufallsdaten, z. B. für die Generierung von Zertifikaten oder kryptografischen Schlüsseln, müssen geeignete sichere Zufallszahlengeneratoren verwendet werden. In BSI-TR-02102-1 wird der Einsatz eines physikalischen und/oder deterministischen Zufallszahlengenerators empfohlen. Da jedoch zu erwarten ist, dass das Datenbanksystem auf einem Server ohne zertifizierte kryptografische Hardware betrieben wird, ist es notwendig, auf einen nicht-physikalischen, nicht-deterministischen Zufallszahlengenerator zurückzugreifen. Unter Linux steht für diesen Zweck die Gerätedatei /dev/random zur Verfügung, die in der Regel eine ausreichende Entropie aufweist. Allerdings muss beim Einsatz von Virtualisierungslösungen die Gefahr von Seitenkanalangriffen in Betracht gezogen werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob ein sicherer Zufallszahlengenerator zur Verfügung steht und systemseitig verwendet wird. Beim Einsatz von /dev/random ist insbesondere darauf zu achten, dass nicht die weniger sichere Alternative /dev/urandom zum Einsatz kommt. Der Unterschied besteht in erster Linie darin, dass /dev/random die Ausgabe von Zufallsdaten blockiert, wenn eine ausreichende Entropie nicht mehr gewährleistet werden kann.

A64 Die Anwendungsdaten müssen verschlüsselt werden.

BSI: B77 | **CIS:** C29 | **STIGs:** S12, S104, S105

Beschreibung Um die Offenlegung und Änderung von Anwendungsdaten im Ruhezustand zu verhindern, müssen diese in verschlüsselter Form gespeichert werden. Dies kann ebenfalls unter Verwendung des symmetrischen AES-Verfahrens und einem sicheren Betriebsmodus erfolgen. Für Letzteres wird in BSI-TR-02102-1 der Einsatz von AES-XTS empfohlen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Verschlüs-

selung der Anwendungsdaten direkt vom Datenbanksystem durchgeführt und aktiviert werden kann.

3.2.10 Verzeichnis- und Dateiberechtigungen

A65 Die Zugriffsrechte auf zur Datenbankanwendung gehörende Verzeichnisse, Dateien und Anwendungen müssen restriktiv vergeben werden.

BSI: B5 | **CIS:** C38, C42 | **STIGs:** S17, S26, S39, S74, S103, S108, S107, S106, S8, S35, S71

Beschreibung Bei der Installation des Datenbanksystems werden verschiedene Verzeichnisse, Dateien und ausführbare Dateien erstellt bzw. gespeichert. Darüber hinaus werden während des Betriebes der Anwendung weitere Daten erzeugt.

Umsetzung Das DBMS muss daraufhin untersucht werden, welche Daten im Rahmen der Installation und durch den Betrieb der Anwendung erzeugt bzw. hinterlegt werden. Zu diesem Zweck stehen die folgenden Verfahren zur Verfügung:

- Der Einsatz des Tools Inotify zur Überwachung des Dateisystems während der Installation.
- Die Analyse des für die Installation herangezogenen Anwendungspakets.
- Die Durchsicht der Anwendungsdokumentation auf relevante Informationen.
- Die Untersuchung der Verzeichnispfade, die unter Linux üblicherweise für die Speicherung von entsprechenden Datentypen verwendet werden (/etc, /lib, /opt, /run, /usr und /var).

Für diese Datenbestände, aus denen sich das gesamte Datenbanksystem zusammensetzt, müssen die Eigentums- und Zugriffsrechte nach dem Least-Privilege- und dem Erforderlichkeitsprinzip zugewiesen werden.

A66 Die Zugriffsrechte auf Protokollierungsdaten müssen restriktiv vergeben werden.

BSI: B65, B66 | **CIS:** C47 | **STIGs:** S2, S3, S64, S65, S66, S67, S68, S69

Beschreibung In Erweiterung der Anforderung A51 muss der Zugriff auf den Inhalt der Protokolldaten nicht nur durch die Autorisierung des Datenbanksystems, sondern auch auf der Ebene des Dateisystems sichergestellt werden. Dabei ist zu beachten, dass der ausführende Systembenutzer bzw. -gruppe nicht imstande sein darf, auf die Protokollierungsdaten zuzugreifen oder diese nachträglich zu verändern.

Umsetzung Das DBMS muss daraufhin untersucht werden, unter welchem Verzeichnis Protokolldaten abgelegt bzw. erzeugt werden (siehe Anforderung A49). Für diese müssen die Eigentums- und Zugriffsrechte nach dem Least-Privilege- und dem Erforderlichkeitsprinzip zugewiesen werden.

A67 Die Zugriffsrechte auf kryptografische Schlüssel müssen restriktiv vergeben werden.

BSI: B41, B45 | **CIS:** C37 | **STIGs:** S23, S88

Beschreibung Wie zuvor bei Anforderung A66 ist diese Anforderung bei korrekter und vollständiger Umsetzung der Anforderung A65 bereits erfüllt. Aufgrund der erheblichen Wichtigkeit des Schutzes aller kryptografischen Schlüssel wurde diese für eine verbesserte Sichtbarkeit nochmals in eine eigene Anforderung ausgelagert.

Umsetzung Das DBMS muss daraufhin untersucht werden, unter welchem Verzeichnis kryptografische Schlüssel gespeichert werden. Dabei ist zu beachten, dass diese nicht ausschließlich in Form von gesonderten Zertifikatsdateien vorliegen, sondern oftmals auch in Konfigurationsdateien enthalten sind. Für diese müssen die Eigentums- und Zugriffsrechte nach dem Least-Privilege- und dem Erforderlichkeitsprinzip zugewiesen werden.

3.2.11 Sicherer Betrieb der Datenbankanwendung

A68 Datenbankspezifische Schutzmechanismen müssen konfiguriert und aktiviert werden.

BSI: B60 | **CIS:** C73, C74

Beschreibung Das Datenbanksystem muss auf besondere Schutzmechanismen überprüft werden. Diese Mechanismen müssen eingesetzt werden, sofern kein vergleichbarer Schutz gegeben ist oder triftige Gründe dagegen sprechen. PostgreSQL enthält beispielsweise die Funktion Row Level Security (RLS), mit der Filter auf Datenbanktabellen angewendet werden können, um Abfragekriterien im Hinblick auf eine Sicherheitsrichtlinie durchzusetzen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob individuelle Schutzmechanismen zur Verfügung stehen und konfiguriert werden können.

A69 Funktionen, die die Ausführung von dynamischem Code verhindern, müssen aktiviert werden.

BSI: B46, B55 | **STIGs:** S9, S41, S27, S110, S111

Beschreibung Die Ausführung von dynamischem Code kann in Datenbanksystemen z. B. durch Code-Injection-Angriffe ausgelöst werden. Bei relationalen Datenbanksystemen ist in diesem Zusammenhang die Kategorie der SQL-Injection-Angriffe sehr bekannt und weit verbreitet. Derartige Angriffe können jedoch auch bei nicht-relationalen Datenbanksystemen durchgeführt werden, da grundsätzlich jede Abfragesprache eine Syntax mit Programmablaufzeichen repräsentiert, die entsprechend umgelenkt und damit attackiert werden kann.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Schutzmechanismen zur Vermeidung von Code-Injection-Angriffen eingesetzt werden können. Typischerweise beinhalten datenbankseitige Mechanismen eine Form der Speicherung der intendierten Datenbankabfrage direkt im Datenbanksystem, sodass lediglich Variablen an diese übergeben werden müssen. Dies hat den Vorteil, dass der Abfragefluss bereits statisch festgelegt ist und eine Umleitung/Verfremdung durch einen Angreifer nicht mehr möglich ist. In der Regel kann dies jedoch nicht im Rahmen einer anwendungsunabhängigen Härtung erfolgen, da derartige hinterlegte Datenbankabfragen individuell und applikationsbezogen sind.

A70 Die Ausführung von Datenbank-Skripten muss deaktiviert werden, oder die Skripte müssen umfassend auf Schwachstellen geprüft werden.

BSI: B76 | **CIS:** C36 | **STIGs:** S81

Beschreibung Einige Datenbanksysteme unterstützen die Verarbeitung von Skripten, wie z. B. JavaScript-Code, für bestimmte benutzerdefinierte Operationen. Die Ausführung solcher datenbankseitiger Skripte stellt ein hohes Risiko dar, da der auszuführende Code Schwachstellen aufweisen kann, die z. B. Code-Injection-Angriffe ermöglichen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die datenbankseitige Ausführung von Skripten unterbunden werden kann. Wenn die Ausführung von Skripten für den Anwendungszweck erforderlich ist, muss zunächst geprüft werden, ob das verfolgte Ziel auch mit anderen Mitteln erreicht werden kann, da das Sicherheitsrisiko in der Regel nicht im Verhältnis dazu steht. Ist dies nicht der Fall, so müssen alle verwendeten Skripte umfassend auf Schwachstellen geprüft werden.

A71 Die verfügbaren Systemressourcen müssen für den Datenbankbetrieb optimiert werden.

BSI: B70 | **CIS:** C35

Beschreibung Die meisten Betriebssysteme, darunter auch Linux, beschränken die Nutzung von Systemressourcen wie Dateideskriptoren, Threads oder auch Netzwerkverbindungen (siehe Anforderung A20) auf der Grundlage des jeweiligen Prozesses oder des ausführenden Benutzers. Diese Beschränkungen sollen Systeminstabilitäten bzw. eine Überbeanspruchung durch einen einzelnen Nutzer verhindern, sind aber im Hinblick auf eine Single-Tenancy-Architektur in der Standardkonfiguration zu niedrig angesetzt. Um alle Systemressourcen bestmöglich zu nutzen und damit die Leistungsfähigkeit zu verbessern, müssen die Grenzwerte angepasst/optimiert werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die betriebssystemseitigen Ressourcenlimits für das Datenbanksystem optimiert sind. Diese können unter /etc/security/limits.conf auf die folgenden empfohlenen Grenzwerte angepasst werden:

- Dateigröße (f): Unbegrenzt
- CPU-Zeit (t): Unbegrenzt
- Virtueller Speicher (v): Unbegrenzt
- Aktive Dateideskriptoren (n): 64 000
- Speichergröße (m): Unbegrenzt
- Aktive Prozesse/Threads (u): 64 000

A72 Die Datenbank muss erfolgreich initialisiert werden.

CIS: C41

Beschreibung Eine fehlerhafte Instanziierung der Datenbank führt zu einem gestörten Betrieb. Ebenso verhält es sich mit Problemen bei der Synchronisation im Cluster-Betrieb. Die Statusinformationen des Datenbanksystems müssen daher im Hinblick auf einen reibungslosen Betrieb bewertet werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Statusinformationen über den gegenwärtigen Betriebszustand abgefragt werden können, um auf Störungen frühzeitig reagieren zu können (vgl. Anforderung A52).

A73 Die Datenbank muss in einen stabilen Zustand übergehen, sollte die Initialisierung fehlschlagen.

STIGs: S7, S101

Beschreibung Im Falle einer Systemstörung oder eines Systemausfalls trägt der Übergang in einen stabilen Zustand dazu bei, den Verlust der Integrität oder der Verfügbarkeit von Daten, z. B. in Form von Datenverlust, zu verhindern. Zu diesem Zweck muss es immer möglich sein, Datenbestände in einen konsistenten Zustand zu überführen, indem Transaktionen abgeschlossen oder rückgängig gemacht werden. Dabei ist auch der Betrieb in einem Cluster zu berücksichtigen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob Methoden oder Betriebsarten zur Verfügung stehen, die bei einer Systemstörung einen Übergang in einen konsistenten Zustand ermöglichen. Im Falle eines Cluster-Betriebs kann dies in Abhängigkeit von der eingesetzten Replikationsmethode auch dazu genutzt werden, eine Systemunterbrechung ganz oder teilweise zu

überbrücken und anschließend die Rückführung in einen stabilen Zustand zu unterstützen.

A74 Die Datenbankanwendung muss unter eigenem Benutzer und eigener Gruppe ausgeführt werden.

CIS: C1

Beschreibung Das Datenbanksystem muss mit einem eigenen Benutzer und einer eigenen Gruppe betrieben werden. Dies verringert die potenziellen Auswirkungen einer Kompromittierung, da z. B. eine Rechteausweitung bei aktuellstem Patch-Stand nicht trivial ist.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob der zur Ausführung verwendete Systembenutzer und -gruppe angepasst werden können. In der Regel muss dazu die Systemd-Unit-Datei des Datenbanksystems unter /etc/systemd/system oder /usr/lib/systemd/system entsprechend angepasst werden.

A75 Die Datenbankanwendung muss mit möglichst geringen Berechtigungen ausgeführt werden.

BSI: B49 | **CIS:** C2, C6, C9, C22, C69

Beschreibung Neben dem Betrieb des Datenbanksystems mit eigenem Benutzer und eigener Gruppe müssen auch die Berechtigungen auf das notwendige Minimum reduziert werden. Im Falle einer Kompromittierung des Datenbanksystems können so die Auswirkungen erheblich verringert werden, indem die Möglichkeit der Ausweitung von Privilegien eingeschränkt und der Umfang eines möglichen Datenzugriffs reduziert wird.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die systemseitige Ausführung ohne root-Rechte erfolgt und ob die dem Systembenutzer zugewiesenen Gruppen ebenfalls dem Least-Privilege- und Erforderlichkeitsprinzip entsprechen. So darf beispielsweise keine Zugehörigkeit zur root- oder sudo/wheel-Systemgruppe bestehen.

A76 Die Datenbankanwendung darf nicht an 0.0.0.0 bzw. [::] gebunden werden.

CIS: C10

Beschreibung Einige Datenbanksysteme binden sowohl die Management- als auch die Anwendungsschnittstellen an die sogenannte Standardroute 0.0.0.0 oder [::]. Dies kann zu einer ungewollten Offenlegung der Schnittstellen gegenüber allen angeschlossenen Netzbereichen führen. Hierdurch kann ein Angreifer in die Lage versetzt werden, aus einem benachbarten, nicht-autorisierten Netzbereich auf das Datenbanksystem zugreifen zu können, wodurch das System möglicherweise kompromittiert oder anderweitig beeinträchtigt werden kann.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Schnittstellen vorzugsweise an den lokalen Hosts oder an einen einzelnen statischen Host gebunden werden können.

A77 Die Datenbankanwendung darf nicht an den Standard-Port gebunden werden.

CIS: C34

Beschreibung Eine Vielzahl von TCP/UDP-Ports wurde von der Internet Assigned Numbers Authority (IANA) im Hinblick auf die Zuordnung zu einem bestimmten Protokoll/Dienst standardisiert. Basierend auf dieser Zuordnung wird beim Aufbau einer Verbindung über ein standardisiertes Protokoll/Dienst immer der Standard-Port verwendet, z. B. Port 80 für HTTP- und Port 443 für HTTPS-Anfragen, sofern nicht ausdrücklich ein anderer Port angegeben wird. Eine Änderung des Standard-Ports kann daher unter Umständen die Identifizierung des Dienstes erschweren und seine Entdeckung verlangsamen. Dies setzt jedoch voraus, dass der Dienst beim Verbindungsaufbau keine für ihn typische Antwort sendet. Insgesamt stellt diese Maßnahme daher keinen wesentlichen Sicherheitsgewinn dar, sondern ist eher dem Bereich der Obfuskation zuzuordnen. Eine Umsetzung sollte dennoch u. a. wegen des sehr geringen Implementierungsaufwands erfolgen.

Umsetzung Das DBMS ist daraufhin zu untersuchen, ob der Standard-Port auf einen beliebigen Wert angepasst werden kann. In Bezug auf die Anforderung A75 ist zu beachten, dass die Bindung eines standardisierten Ports (0 – 1023) root-Rechte voraussetzt.

A78 Die Management-Schnittstelle muss sich in einem dedizierten Netzwerksegment befinden und der Zugriff begrenzt werden.

STIGs: S97

Beschreibung Die Management-Schnittstelle dient der Konfiguration des Datenbanksystems und ermöglicht sowohl den Zugriff als auch das Exportieren von Daten sowie eine Reihe weiterer Funktionen. Wird der Zugriff auf diese Schnittstelle nicht eingeschränkt und damit im ungünstigsten Fall allen Netzwerkteilnehmern zugänglich gemacht, besteht eine hohe Angriffsgefahr. Angesichts des umfangreichen Zugriffs und des daraus entstehenden Sicherheitsrisikos muss die Management-Schnittstelle in ein gesondertes geschütztes Netzsegment ausgelagert werden.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob die Management-Schnittstelle in ein separates Netzwerksegment ausgelagert oder der Zugriff an den lokalen Host gebunden werden kann.

A79 Die Systemd-Dienstdateien müssen aktiviert werden.

CIS: C40

Beschreibung Durch die Aktivierung der zugehörigen Systemd-Dienstdateien wird sichergestellt, dass das Datenbanksystem bei einer Zustandsveränderung automatisch gestartet wird. Hierzu kann ein Neustart des Betriebssystems, die Installation von Aktualisierungspaketen oder auch das Auftreten einer Störung gehören.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob ein automatischer Start der Anwendung über Systemd-Dienstdateien erfolgen kann. In der Regel muss dazu die Systemd-Unit-Datei des Datenbanksystems unter /etc/-systemd/system oder /usr/lib/systemd/system durch den Systemd-System- und Sitzungsmanager aktiviert werden.

A80 Die Systemzeit muss über das Network Time Protocol synchronisiert werden.

BSI: B63, B64 | **CIS:** C3

Beschreibung Die Genauigkeit der Systemzeit des Datenbanksystems ist entscheidend für den Betrieb eines Clusters. Einige Datenbanksysteme verwenden

Zeitstempel, um über die aktuellsten Datensätze für die Steuerung der Synchronisation und Replikation zu entscheiden. Daher kann es bei Diskrepanzen der Systemzeit zwischen den Cluster-Teilnehmern zu Synchronisationsfehlern oder Datenverlusten kommen. Darüber hinaus ist die korrekte Systemzeit für die Auditierung und Protokollierung von großer Bedeutung.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine kontinuierliche Zeitsynchronisation auf der Seite der Anwendung oder des Betriebssystems über das Network Time Protocol (NTP) möglich ist.

3.2.12 Backup und Replikation

A81 Es müssen regelmäßige Systemsicherungen des Datenbanksystems durchgeführt werden.

BSI: B68, B69 | **CIS:** C83, C84, C85

Beschreibung Für den Fall eines Datenverlustes, etwa infolge von Synchronisationsfehlern, Systemausfällen oder einer erfolgreichen Manipulation durch einen Angreifer, sind regelmäßige Systemsicherungen als vorbeugende Maßnahme durchzuführen. Bei einem Cluster-Betrieb ist abhängig von der Replikationsmethode darauf zu achten, dass eine Sicherung des Master-Datenbanksystems möglicherweise nicht ausreicht, um alle vorhandenen Daten in die Sicherung einzubeziehen.

Umsetzung Das DBMS muss daraufhin untersucht werden, ob eine automatisierte regelmäßige Datensicherung eingerichtet werden kann. Dabei können nicht nur die Daten der Datenbanken von Interesse sein, sondern unter anderem auch die Konfigurationsdateien.

A82 Für die Durchführung von Systemsicherungen muss ein eigenständiger Benutzer verwendet werden.

CIS: C81

Beschreibung Die Durchführung von Sicherungen muss durch einen eigenen Benutzer bzw. Benutzergruppe/ACL durchgeführt werden. Dies verringert die potenziellen Auswirkungen einer Kompromittierung, da z. B. eine Rechteausweitung bei aktuellstem Patch-Stand nicht trivial ist.

Umsetzung Das DBMS muss daraufhin untersucht werden, welche Methoden zur Durchführung einer Datensicherung zur Verfügung stehen. Wenn die automatisierte Durchführung direkt im Datenbanksystem konfiguriert werden kann, ist es sinnvoll, dafür eine eigene Benutzergruppe/ACL anzulegen. Gemäß dem Least-Privilege-Prinzip ist ein Lesezugriff in der Regel ausreichend, um die Aufgabe zu erfüllen. Erfolgt die Sicherung auf Betriebssystemebene, sollte der Zugriff auf die Sicherheitskopien für den ausführenden Systembenutzer und die -gruppe unzugänglich sein (vgl. Anforderung A74). Darüber hinaus muss generell sichergestellt werden, dass Sicherungen nicht nachträglich bearbeitet werden können.

4 Anwendung des Vorgehensmodells

In diesem Kapitel wird das im vorherigen Kapitel 3 entwickelte Vorgehensmodell am Beispiel von zwei Datenbanksystemen angewendet. Dazu wird für jede Anforderung geprüft, ob sie aus technischer Sicht für das entsprechende DBMS umgesetzt werden kann. Darüber hinaus wird zur Vorbereitung der Überführung in InSpec-Anwendungscode die konkret auszuführende technische Aktion erfasst, beispielsweise der numerische Abgleich von Dateiberechtigungen oder eine auszuführende Datenbankabfrage.

4.1 Ableitung von Maßnahmen für Neo4j und Redis

Für die Ableitung von Maßnahmen wurde das Datenbanksystem Neo4j in der Version 5.1.0 und Redis in der Version 7.0.4 in den jeweiligen Community-Editionen ausgewählt. Neo4j wurde einerseits wegen seiner Popularität im Bereich der Graphdatenbanken [37] sowie aufgrund der erst am 6. Oktober 2022 veröffentlichten Version 5.x ausgewählt [86]. Redis hingegen wurde neben seiner Popularität [34] auch deshalb ausgewählt, um im Rahmen der Auswertung im folgenden Kapitel 5 untersuchen zu können, ob im Vergleich zu „Redis Enterprise 6.x STIG Version 1, Release 1“ unter Berücksichtigung der Versions- und Editionsunterschiede weitere Maßnahmen identifiziert wurden. Als unterliegendes Betriebssystem kommt Ubuntu 22.10 zum Einsatz.

Die Grundlage/Quelle für die Identifizierung aller Maßnahmen bilden die jeweiligen Produktdokumentation sowie die mitgelieferten Standardkonfigurationen [87, 88].

4.1.1 Grundprinzipien des Vorgehensmodells

A1 Die Standardwerte aller sicherheitsrelevanten Konfigurationsparameter müssen explizit festgelegt werden.

Neo4j Es konnten mehrere sicherheitsrelevante Konfigurationsoptionen für Neo4j identifiziert werden, die keiner der nachfolgenden Anforderungen zu-

geordnet werden konnten. Diese können in der Datei /etc/neo4j/neo4j.conf konfiguriert werden.

1. Der Zugriff auf Java Management Extensions, die eine der potenziellen Angriffsmöglichkeiten im Zusammenhang mit der Log4Shell-Schwachstelle (CVE-2021-44228) darstellten, sollte wie folgt deaktiviert werden:

```
# JMX-Endpoint deaktivieren (Standardwert false)
server.jvm.additional=-Dlog4j2.disable.jmx=true
```

2. Das OCSP-Stapling zur Überprüfung des Widerrufsstatus Zertifikaten kann wie folgt aktiviert werden:

```
# OCSP-Stapling für Bolt aktivieren (Standardwert false)
server.bolt.ocsp_stapling_enabled=true
```

3. HTTP Strict-Transport-Security kann wie folgt aktiviert werden:

```
# HSTS für HTTPS aktivieren (Standardwert <leer>)
dbms.security.http_strict_transport_security=max-age=15768000
```

Redis Es konnte eine sicherheitsrelevante Konfigurationsoption für Redis identifiziert werden, die keiner der nachfolgenden Anforderungen zugeordnet werden konnte.

Standardmäßig ändert Redis den Prozesstitel, um einige Laufzeitinformationen bereitzustellen. Dies kann wie folgt in der Datei /etc/redis/redis.conf deaktiviert werden:

```
# Laufzeitinformationen im Prozesstitel deaktivieren (Strandardwert
↪ yes)
set-proc-title no
```

A2 Nicht benötigte Plug-ins/Software-Erweiterungen und Funktionen müssen deinstalliert oder deaktiviert werden.

Neo4j Neo4j wird in der Community-Edition ohne vorinstallierte Plug-ins/Software-Erweiterungen ausgeliefert/installiert. Dies kann in der Datei /etc/-

neo4j/neo4j.conf nachgeprüft werden, indem kontrolliert wird, ob die folgenden Zeilen auskommentiert oder entfernt wurden:

```
# Laden aller Plug-ins deaktivieren (Standardwert <auskommentiert>)
#server.unmanaged_extension_classes=
#dbms.security.procedures.unrestricted=
#dbms.security.procedures.allowlist=
```

Redis Redis wird in der Community-Edition ohne vorinstallierte Plug-ins/ Software-Erweiterungen ausgeliefert/installiert. Dies kann mit dem folgenden Befehl kontrolliert werden:

```
# Alle geladenen Module anzeigen (Redis-CLI)
MODULE LIST
```

Der zu erwartende Rückgabewert ist `(empty array)`. Darüber hinaus muss in der Datei /etc/redis/redis.conf nachgeprüft werden, ob alle Zeilen mit dem folgenden Präfix auskommentiert oder entfernt wurden:

```
# Laden aller Plug-ins deaktivieren (Standardwert <auskommentiert>)
#loadmodule /etc/redis/example-module.so
# Laden aller Zusatzkonfigurationen deaktivieren (Standardwert
↪ <auskommentiert>)
#include /path/to/other.conf
```

4.1.2 Installation und Updates

A3 Der Aktualisierungsmechanismus muss sicher konfiguriert werden.

Für Neo4j und Redis ist kein integrierter Aktualisierungsmechanismus vorhanden. Beide verwenden das vom Betriebssystem bereitgestellte Paketverwaltungssystem APT. Für dessen sicheren Betrieb muss sichergestellt werden, dass

- die zum Betriebssystem gehörenden GPG-Schlüssel unter /etc/apt/trusted.gpg.d/ gespeichert sind, da deren Verwaltung durch `apt-key` nicht mehr zeitgemäß und weniger sicher ist.

- alle Verzeichnisse und Dateien unter /etc/apt/ (rekursiv) dem Systembenutzer und der -gruppe root gehören.
- alle Verzeichnisse unter /etc/apt/ (rekursiv) nur Schreibrechte für den Eigentümer aufweisen, d. h. in oktaler Notation maximal 0755.
- alle Dateien unter /etc/apt/ (rekursiv) nur Schreibrechte für den Eigentümer und grundsätzlich keine Ausführungsrechte aufweisen, d. h. in oktaler Notation maximal 0644.

A4 Die Herkunft der Software-Installations- und -Aktualisierungspakete aus vertrauenswürdigen Quellen muss gewährleistet werden.

Für die Konfiguration von Drittanbieter-APT-Repositories muss für jeden Anbieter eine separate Datei unter /etc/apt/sources.list.d/ erstellt werden. Um Cross-Signing-Operationen zu verhindern, wird mit dem Argument `signed-by` für die Umsetzung der Anforderung A6 bereits sichergestellt, dass die Signaturprüfung mit einem bestimmten GPG-Schlüssel durch `apt-secure` erzwungen wird, und nicht mit allen in `apt-key` konfigurierten vertrauenswürdigen Schlüsseln erfolgen kann.

Neo4j Der Bezug von offiziellen Installationspaketen für Neo4j von `https://debian.neo4j.com` kann mit dem folgenden Befehl eingerichtet werden:

```
1 # Offizielles Neo4j-APT-Repository konfigurieren
2 echo "deb [signed-by=/etc/apt/keyrings/neotechnology-keyring.gpg]
  ↪ https://debian.neo4j.com stable 5" | tee
  ↪ /etc/apt/sources.list.d/neo4j.list
```

Redis Der Bezug von offiziellen Installationspaketen für Redis von `https://packages.redis.io` kann mit dem folgenden Befehl eingerichtet werden:

```
1 # Offizielles Redis-APT-Repository konfigurieren
2 echo "deb [signed-by=/etc/apt/keyrings/redis-archive-keyring.gpg]
  ↪ https://packages.redis.io/deb $(lsb_release -cs) main" | tee
  ↪ /etc/apt/sources.list.d/redis.list
```

A5 Die Herkunft von Plug-ins/Software-Erweiterungen aus vertrauenswürdigen Quellen muss gewährleistet werden.

Neo4j Neo4j verfügt in der Community-Edition nicht über einen integrier-

ten Online-Marktplatz oder ähnliches, um Plug-ins/Software-Erweiterungen zu installieren. Stattdessen werden diese in Form einer `.jar`-Datei bereitgestellt. Eine Übersicht über die offiziellen Software-Erweiterungen des Herstellers ist unter `https://neo4j.com/download-center/#add-on` verfügbar und kann als Orientierungshilfe herangezogen werden. Eine Beschränkung darauf ist allerdings technisch nicht durchsetzbar, sondern muss durch organisatorische, prozessuale Maßnahmen realisiert werden.

Redis Redis verfügt in der Community-Edition nicht über einen integrierten Online-Marktplatz oder ähnliches, um Plug-ins/Software-Erweiterungen zu installieren. Stattdessen werden diese je nach Entwickler entweder als gemeinsam genutzte Bibliothek (englisch „Shared Library“) in Form einer `.so`-Datei bereitgestellt oder müssen vom Administrator zunächst selbst kompiliert werden. Eine Übersicht mit den vom Hersteller empfohlenen Software-Erweiterungen ist hingegen unter `https://redis.io/resources/modules/` verfügbar und kann als Orientierungshilfe herangezogen werden. Eine Beschränkung darauf ist allerdings technisch nicht durchsetzbar, sondern muss durch organisatorische, prozessuale Maßnahmen realisiert werden.

A6 Die Integrität der Software-Installations- und -Aktualisierungspakete muss verifiziert werden.

Um die Integrität der Software-Installations- und Aktualisierungspakete aus den Drittanbieter-APT-Repositories zu gewährleisten, müssen deren GPG-Schlüssel unter /etc/apt/keyrings/ gespeichert werden, um eine Signaturprüfung der Pakete zu ermöglichen.

Neo4j Für Neo4j kann dies mit dem folgenden Befehl umgesetzt werden:

```
1 # Offiziellen Neo4j-GPG-Schlüssel einrichten
2 curl -fsSL https://debian.neo4j.com/neotechnology.gpg.key | gpg
  ↪  --dearmor -o /etc/apt/keyrings/neotechnology-keyring.gpg
```

Redis Für Redis kann dies mit dem folgenden Befehl umgesetzt werden:

```
1 # Offiziellen Redis-GPG-Schlüssel einrichten
2 curl -fsSL https://packages.redis.io/gpg | gpg --dearmor -o
  ↪  /etc/apt/keyrings/redis-archive-keyring.gpg
```

A7 Die Integrität der Plug-ins/Software-Erweiterungen muss verifiziert werden.

Für Neo4j und Redis gibt es keinen standardisierten Bereitstellungsmechanismus, der beispielsweise eine Signatur- oder Prüfsummenverifizierung für die Installation von Plug-ins/Software-Erweiterungen vorsieht. Daher muss darauf vertraut werden, dass die Sicherheitsmaßnahmen des Entwicklers hinreichend wirksam sind.

Neo4j Für Neo4j sind Prüfsummen in Form von SHA-256 für die offiziellen Plug-ins verfügbar und die entsprechenden Pakete werden mit dem öffentlichen PGP-Schlüssel ID `1BD0DB31` signiert (vgl. Umsetzung der Anforderung A6).

Redis Für Redis werden Plug-ins in Form von `.so`-Dateien oder ausschließlich als Quellcode bereitgestellt. Für Ersteres stehen für die offiziellen Plug-ins keine Prüfsummen zur Verfügung und werden auch von Drittanbietern nur in Einzelfällen bereitgestellt.

A8 Die Version der Software und der Plug-ins/Software-Erweiterungen müssen vom Hersteller unterstützt werden.

Für Neo4j und Redis kann mithilfe des Paketverwaltungssystems APT sowohl die installierte als auch eine ggf. verfügbare aktuellere Paketversion wie folgt ermittelt werden:

```
1 # Paketquellen aktualisieren und auf neuere Version prüfen
2 apt update; apt-cache policy neo4j | awk '/Installed: / ||
  ↪ /Candidate: /{print $NF}' | uniq -c
3 apt update; apt-cache policy redis-server | awk '/Installed: / ||
  ↪ /Candidate: /{print $NF}' | uniq -c
```

Der Wert `Installed:` aus dem Rückgabewert gibt die installierte Version an und der Wert `Candidate:` die aktuellste verfügbare Version, sofern vorhanden. Anhand dieser Daten kann dann festgestellt werden, ob in Abhängigkeit zum Veröffentlichungszyklus des Datenbanksystems (siehe nachfolgend) die vorliegende Version unterstützt wird und aktuell ist. Eine korrekte Auswertung erfordert allerdings eine Verbindung zu einem tagesaktuellen APT-Paketverwaltungsserver. Als Ausweichmöglichkeit kann daher auch ein Abgleich mit einer statischen Liste unterstützter Versionen, die kontinuierlich aktualisiert werden muss, genutzt werden.

Neo4j Für Neo4j hat sich mit der Veröffentlichung von Version 5 das Vorgehen im Veröffentlichungszyklus für alle kommenden 5.x-Versionen grundlegend geändert. Jede Hauptversion wird unterstützt, bis die nächste Nebenversion veröffentlicht wird. Neuere Nebenversionen ersetzen wiederum direkt die vorherige Nebenversion, und so weiter und so fort. Werden Unterversionen von Nebenversionen in Form von sogenannten Hotfixes veröffentlicht, um z. B. dringende Sicherheitslücken zu schließen, ersetzen diese ebenfalls direkt die aktuelle Nebenversion. Somit kann es immer nur eine unterstützte Neo4j 5.x Version geben. Für Neo4j 4.x bleibt die alte Vorgehensweise bestehen, die aufgrund der geringen noch verbleibenden Unterstützung nicht weiter thematisiert wird. Mit Stand vom 28. November 2022 werden folgende Versionen vom Hersteller unterstützt:

- 5.1.0 (unterstützt bis zur Veröffentlichung von Version 5.2.0 oder eines Hotfixes),
- 4.4.12 (unterstützt bis einschl. 01.12.2024),
- 4.3.20 (unterstützt bis einschl. 16.12.2022).

Redis Für Redis sieht der Veröffentlichungszyklus vor, dass pro Jahr eine neue Hauptversion veröffentlicht wird. Nach 6 Monaten folgt die Veröffentlichung einer weiteren Nebenversion. Unterstützt werden vom Hersteller immer die aktuellste Hauptversion, die aktuellste vorherige Nebenversion und die aktuellste vorherige Hauptversion. Somit werden vom Hersteller immer drei Versionen parallel unterstützt, und zwar mit Stand vom 28. November 2022 die Folgenden:

- 7.0.5 (Hauptversion 7.0),
- 6.2.7 (Nebenversion 6.2),
- 6.0.16 (vorherige Hauptversion 6.0).

A9 Die Installation von Aktualisierungspaketen muss zeitnah erfolgen.

Im Rahmen der Umsetzung der Anforderung A8 wurden bereits alle technischen Maßnahmen zur Erfüllung dieser Anforderung für Neo4j und Redis umgesetzt. Weitere Sicherheitsmaßnahmen können nur durch organisatorische, prozessuale Maßnahmen realisiert werden.

4.1.3 Authentifizierung

A10 Die Authentifizierung muss konfiguriert und aktiv sein.

Neo4j Für Neo4j kann die Authentifizierung in der Datei /etc/neo4j/neo4j.conf wie folgt aktiviert werden:

```
1 # Authentifizierung aktivieren (Standardwert false)
2 dbms.security.auth_enabled=true
```

Redis Für Redis kann die Standard-Authentifizierung in der Datei /etc/redis/redis.conf wie folgt aktiviert werden:

```
1 # Authentifizierung aktivieren (Standardwert <auskommentiert>)
2 requirepass examplepassword
```

Diese Konfigurationsoption stellt jedoch nur eine Kompatibilitätsschicht zum neuen ACL-System ab Redis Version 6 dar. Durch die zusätzliche Verwendung dieser Option wird nur sichergestellt, dass der Standardbenutzer das angegebene Passwort für die Authentifizierung verwendet, solange keine ACL-Benutzerdatei vorhanden ist (siehe Umsetzung der Anforderung A11).

A11 Sind mehrere Authentifizierungsmechanismen verfügbar, ist das sicherste Verfahren zu verwenden.

Neo4j Neo4j verfügt in der Community-Edition über keine zusätzlichen Authentifizierungsmechanismen, wie z. B. LDAP oder Kerberos.

Redis Redis unterstützt die Verwendung einer externen ACL-Benutzerdatei, um die Verwaltung von Benutzerkonten wie folgt auszulagern:

```
1 # ACL-Datei verwenden (Standardwert <auskommentiert>)
2 aclfile /etc/redis/users.acl
```

A12 Die Authentifizierung aller Teilnehmer in einem Cluster muss konfiguriert und aktiv sein.

Neo4j und Redis ermöglichen in der Community-Edition keinen Betrieb von Clustern.

A13 Die Authentifizierung muss an sämtlichen Schnittstellen/Interfaces erfolgen.

Neo4j Neo4j verfügt in der Community-Edition unter Berücksichtigung der hier angestrebten gehärteten Konfiguration über zwei Schnittstellen. Die Bolt-Schnittstelle wird für den Zugriff der angebundenen Anwendungen verwendet und die HTTPS-Schnittstelle dient der Verwaltung des Datenbanksystems. Durch die Umsetzung der Anforderung A10 ist in beiden Fällen die Authentifizierung immer aktiv.

Redis Redis verfügt in der Community-Edition unter Berücksichtigung der hier angestrebten gehärteten Konfiguration nur über eine Schnittstelle. Durch die Umsetzung der Anforderung A10 ist für diese die Authentifizierung immer aktiv.

A14 Die Authentifizierung muss, wenn möglich, mehrere Authentifizierungsmerkmale umfassen (Multi-Faktor-Authentifizierung).

Neo4j Für Neo4j kann eine Multi-Faktor-Authentifizierung in Form einer Client-seitigen Zertifikatsauthentifizierung in der Datei /etc/neo4j/neo4j.conf wie folgt konfiguriert werden:

```
1 # HTTPS Client-seitige Zertifikatsauthentifizierung aktivieren
  ↪ (Standardwert NONE)
2 dbms.ssl.policy.https.client_auth=REQUIRE
3 # Bolt Client-seitige Zertifikatsauthentifizierung aktivieren
  ↪ (Standardwert NONE)
4 dbms.ssl.policy.bolt.client_auth=REQUIRE
```

Redis Für Redis kann eine Multi-Faktor-Authentifizierung in Form einer Client-seitigen Zertifikatsauthentifizierung in der Datei /etc/redis/redis.conf wie folgt konfiguriert werden:

```
1 # Client-seitige Zertifikatsauthentifizierung aktivieren
  ↪ (Standardwert yes)
2 tls-auth-clients yes
```

A15 Sitzungskennungen müssen zufällig erzeugt werden und dürfen kein vorhersehbares Schema aufweisen.

Neo4j und Redis verwenden Sitzungskennungen, um die Authentifizierung einer aktiven Verbindung aufrechtzuerhalten. Für beide Datenbanksysteme konnte kein vorhersehbares Schema ermittelt werden, sodass deren Standard-Generierung als ausreichend sicher angesehen werden kann. Allerdings können diese auch nicht benutzerdefiniert eingestellt oder anderweitig konfiguriert werden.

A16 Fehlgeschlagene Authentifizierungen dürfen nicht zur Durchführung von Angriffen interpretiert werden können.

Neo4j Neo4j gibt im Falle eines fehlgeschlagenen Authentifizierungsversuchs keine Informationen zurück, die zur Durchführung von Angriffen interpretiert werden können. Der erwartete Rückgabewert im Falle einer fehlgeschlagenen Authentifizierung über HTTPS und Bolt entspricht:

```
1 # HTTPS-Authentifizierungsfehler (HTTP 401 Unauthorized)
2 Neo.ClientError.Security.Unauthorized: The client is unauthorized due
  ↪  to authentication failure.
3 # Bolt-Authentifizierungsfehler
4 $ cypher-shell -u neo4j -p wrongpassword
5 The client is unauthorized due to authentication failure.
6 $ cypher-shell -u nonexistent -p examplepassword
7 The client is unauthorized due to authentication failure.
```

Dieses Verhalten kann nicht weiter angepasst werden.

Redis Redis gibt im Falle eines fehlgeschlagenen Authentifizierungsversuchs keine Informationen zurück, die zur Durchführung von Angriffen interpretiert werden können. Der erwartete Rückgabewert im Falle einer fehlgeschlagenen Authentifizierung entspricht:

```
1 # Redis-CLI Authentifizierungsfehler
2 127.0.0.1:6379> AUTH default wrongpassword
3 (error) WRONGPASS invalid username-password pair or user is disabled.
4 127.0.0.1:6379> AUTH nonexistent examplepassword
5 (error) WRONGPASS invalid username-password pair or user is disabled.
```

Dieses Verhalten kann nicht weiter angepasst werden.

A17 Wenn mehrere Authentifizierungsversuche fehlschlagen, müssen Trigger definiert werden, um weitere Versuche zu verzögern.

Neo4j Für Neo4j kann die temporäre Sperrung von Benutzerkonten in der Datei /etc/neo4j/neo4j.conf wie folgt aktiviert werden:

```
1 # Dauer der temporären Sperrung konfigurieren (Standardwert 5s)
2 dbms.security.auth_lock_time=5s
3 # Anzahl der zulässigen Authentifizierungsversuche konfigurieren
  ↪ (Standardwert 3)
4 dbms.security.auth_max_failed_attempts=3
```

Redis Redis verfügt in der Community-Edition nicht über eine integrierte Funktion zur Verzögerung von Authentifizierungsversuchen, wenn eine festgelegte Anzahl von Anmeldungen innerhalb einer vordefinierten Zeitspanne fehlschlagen. Redis schreibt über dieses Problem in seiner Dokumentation [89]: „Aufgrund der hohen Leistungsfähigkeit von Redis ist es möglich, viele Passwörter in sehr kurzer Zeit parallel auszuprobieren. Stellen Sie also sicher, dass Sie ein starkes und sehr langes Passwort generieren, damit dieser Angriff nicht durchführbar ist […]“.

A18 Wenn mehrere Authentifizierungsversuche fehlschlagen, müssen Trigger definiert werden, um aktive Sitzungen zu beenden oder Benutzer zu sperren.

Neo4j und Redis verfügen in der Community-Edition nicht über eine integrierte Funktion, um Benutzerkonten dauerhaft zu sperren oder aktive Sitzungen zu beenden, wenn eine bestimmte Anzahl von Anmeldungen innerhalb einer vordefinierten Zeitspanne fehlschlagen.

A19 Die Gesamtdauer eines Anmeldeversuchs muss begrenzt werden.

Neo4j und Redis bieten keine Möglichkeit, die Gesamtdauer eines Anmeldeversuchs zu begrenzen.

A20 Die Anzahl der gleichzeitigen Verbindungen zur Datenbank muss begrenzt werden.

Neo4j Neo4j verfügt in der Community-Edition nicht über eine Möglichkeit, die Anzahl der gleichzeitigen Verbindungen zur Datenbank zu begrenzen.

Redis In Redis kann die Anzahl der gleichzeitigen Verbindungen zur Daten-

bank nicht begrenzt werden. Stattdessen kann jedoch der Wert von TCP-Keepalive zur Erkennung inaktiver Verbindungen und der Timeout für den TLS-Session-Cache in der Datei /etc/redis/redis.conf wie folgt konfiguriert werden:

```
# TCP-Keepalive konfigurieren (Standardwert 300)
tcp-keepalive 300
# Timeout für TLS-Session-Cache konfigurieren (Standardwert 300)
tcp-keepalive 300
```

A21 Die Anzahl der parallel aktiven Sitzungen pro Benutzer muss begrenzt werden.

Neo4j In Neo4j kann die Anzahl der parallel aktiven Sitzungen in der Community-Edition nicht begrenzt werden.

Redis In Redis kann die Anzahl der parallel aktiven Sitzungen für das gesamte Datenbanksystem begrenzt werden, jedoch nicht pro Benutzer. Dies kann in der Datei /etc/redis/redis.conf wie folgt konfiguriert werden:

```
# Maximale Client-Verbindungen begrenzen (Standardwert 10000)
maxclients 60000
```

A22 Die Zwischenspeicherung von Authentifizierungsdaten muss deaktiviert werden.

Neo4j In Neo4j können Authentifizierungsdaten für die HTTPS-Management-Schnittstelle im Browser zwischengespeichert werden. Dies kann in der Datei /etc/neo4j/neo4j.conf wie folgt deaktiviert werden:

```
# Zwischenspeicherung von Zugangsdaten deaktivieren (Standardwert
↪ true)
browser.retain_connection_credentials=false
```

Darüber hinaus kann eine Sitzungszeitüberschreitung ebenfalls in der Datei /etc/neo4j/neo4j.conf wie folgt konfiguriert werden:

```
# Sitzungszeitüberschreitung konfigurieren (Standardwert 0s)
browser.credential_timeout=30s
```

Redis In Redis kann eine Sitzungszeitüberschreitung in der Datei /etc/redis/redis.conf wie folgt konfiguriert werden:

```
# Sitzungszeitüberschreitung konfigurieren (Standardwert 0)
timeout 30
```

4.1.4 Autorisierung

A23 Die Autorisierung muss konfiguriert und aktiv sein.

In Neo4j und Redis wird die Autorisierung automatisch durch die Einrichtung der Authentifizierung als Teil der Umsetzung der Anforderungen A10 und A11 aktiviert.

A24 Jeder Benutzer muss einer Berechtigungsgruppe/Access Control List zugewiesen sein.

Neo4j In der Community-Edition von Neo4j wird jeder Benutzer einer ACL mit vordefinierten Standardberechtigungen zugewiesen. Diese Vorgehensweise kann nicht individuell angepasst oder anderweitig beeinflusst werden (siehe Umsetzung der Anforderung A32).

Redis Redis verfügt in der Community-Edition nicht über eine Funktionalität zur Erstellung von ACL in Form einer Zuweisung von Benutzern zu einer oder mehreren Berechtigungsgruppen mit vorkonfigurierten Berechtigungen. Stattdessen muss jedem Benutzer auf individueller Basis der Zugriff auf:

- Befehle oder Gruppen von Befehlen,
- Schlüsselwerte auf der Grundlage eines bestimmten Präfixes oder
- Kanäle des Pub/Sub-Nachrichtensystems auf der Grundlage eines bestimmten Präfixes

ausdrücklich gestattet oder verweigert werden.

A25 Benutzerkonten, die über einen längeren Zeitraum inaktiv sind, müssen deaktiviert werden.

Neo4j und Redis verfügen über keine integrierte Funktionalität zur automatischen Deaktivierung von Benutzerkonten nach einem vordefinierten Inaktivitätszeitraum.

A26 Benutzerkonten und -gruppen, die deaktiviert sind/nicht verwendet werden, müssen gelöscht werden.

Neo4j und Redis verfügen über keine integrierte Funktionalität zur automatischen Löschung von deaktivierten Benutzerkonten nach einem vordefinierten Zeitraum.

A27 Die vordefinierten Benutzerrollen sind so weit wie möglich zu verwenden.

Neo4j In der Community-Edition von Neo4j wird jeder Benutzer einer ACL mit vordefinierten Standardberechtigungen zugewiesen. Diese Vorgehensweise kann nicht individuell angepasst oder anderweitig beeinflusst werden (siehe Umsetzung der Anforderung A32).

Redis Redis verfügt in der Community-Edition nicht über eine Funktionalität zur Erstellung von ACL im klassischen Sinne (siehe Umsetzung der Anforderung A24).

A28 Die vordefinierten Benutzerrollen sind auf ihre Vereinbarkeit im Hinblick auf alle Anforderungen zu prüfen.

Neo4j und Redis weisen in ihren vordefinierten Benutzerrollen umfangreiche Berechtigungen auf, die nicht mit allen Anforderungen des Vorgehensmodells zu vereinbaren sind, insbesondere im Hinblick auf eine rollenbasierte Zugriffskontrolle (siehe Umsetzung der Anforderung A31) sowie der Vergabe nach dem Least-Privilege- und Erforderlichkeitsprinzip (siehe Umsetzung der Anforderung A32).

Neo4j In der Community-Edition von Neo4j wird jeder Benutzer einer ACL mit vordefinierten Standardberechtigungen zugewiesen, die nicht angepasst werden können.

Redis In Redis verfügt die Standard-Benutzerkennung `default` über die folgenden Berechtigungen:

```
1 # ACL der Standard-Benutzerkennung
2 user default on nopass ~* &* +@all
```

Die aufgeführten Argumente haben die folgende Bedeutung:

on Aktivierung des Benutzers.

nopass Deaktivierung der Authentifizierung für den Benutzer.

~* Zugriff auf alle Schlüsselwerte ohne Limitierung durch einen Präfix.

&* Zugriff auf alle Kanäle des Pub/Sub-Nachrichtensystems.

+@all Zugriff auf alle Befehle.

Diese weitreichenden Berechtigungen, insbesondere vor dem Hintergrund der deaktivierten Authentifizierung, sind unter Berücksichtigung der hier angestrebten gehärteten Konfiguration nicht mit allen Anforderungen vereinbar, sodass im Rahmen der Umsetzung der Anforderung A34 die Standard-Redis-Benutzerkennung deaktiviert wird.

A29 Die gleiche Benutzerkennung darf nicht von mehreren Personen oder Diensten verwendet werden.

Neo4j und Redis unterstützen in der Community-Edition keine Zugriffsbeschränkung auf Basis der zugreifenden IP-Adresse bezogen auf einen bestimmten Nutzer oder eine Datenbank. Dies lässt sich nur auf der Ebene des Datenbanksystems durch die IP-Bindung realisieren, sodass die Nutzung der gleichen Benutzerkennung durch mehrere Personen oder Dienste nicht ausgeschlossen oder auch nur teilweise erschwert werden kann.

A30 Die gleiche Benutzerkennung darf nicht für den Zugriff auf mehrere Datenbanken verwendet werden.

Neo4j In der Community-Edition von Neo4j wird jeder Benutzer einer ACL mit vordefinierten Standardberechtigungen zugewiesen. Diese Vorgehensweise kann nicht individuell angepasst oder anderweitig beeinflusst werden (siehe Umsetzung der Anforderung A32).

Redis In Redis wird aufgrund des Schlüssel-Wert-Zugriffssystems keine klassische Unterteilung in Datenbanken vorgenommen. Stattdessen wird der Zugriff auf bestimmte Schlüsselwerte oder Kanäle des Pub/Sub-Nachrichtensystems durch Präfixe begrenzt. Um die Konformität mit dieser Anforderung zu überprüfen, können daher die Zugriffspräfixe beider Systeme wie folgt auf Duplikate überprüft werden:

```
# Suche nach Duplikaten von Schlüsselpräfixen
grep -oP '(?<=~)\S+?(?=\*)' /etc/redis/users.acl | uniq -c
# Suche nach Duplikaten von Kanalpräfixen
grep -oP '(?<=&)\S+?(?=\*)' /etc/redis/users.acl | uniq -c
```

Der erwartete Rückgabewert für den Fall, dass zwei Duplikate vorliegen, entspricht folgendem Muster:

```
# Muster-Ausgabe von uniq -c für Schlüsselpräfix-Duplikate
2    examplekey:
# Muster-Ausgabe von uniq -c für Kanalpräfix-Duplikate
2    examplechannel:
```

A31 Eine rollenbasierte Zugriffskontrolle zur Trennung von Benutzer- und Datenbankverwaltungsfunktionen muss umgesetzt werden.

Neo4j In der Community-Edition von Neo4j wird jeder Benutzer einer ACL mit vordefinierten Standardberechtigungen zugewiesen. Diese Vorgehensweise kann nicht individuell angepasst oder anderweitig beeinflusst werden (siehe Umsetzung der Anforderung A32).

Redis In Redis kann dies in der ACL-Datei /etc/redis/users.acl durch die Definition eines administrativen Benutzers und eines Standardbenutzers mit minimalen Zugriffsrechten wie folgt realisiert werden:

```
# Administratives Benutzerkonto konfigurieren
user admin on ~* &* +@all
↪  #749f09bade8aca755660eeb17792da880218d4fbdc4e25fbec279d7fe9f65d70
# Minimales Benutzerkonto konfigurieren
user minimal on resetkeys ~minimal:* resetchannels &minimal:* +@all
↪  -@admin -@dangerous -@scripting
↪  #21adaff8e0b936c51ed239be3935addd85f8e1c0f914dd087d6da10f2d956aab
```

Redis interpretiert die Berechtigungsanweisungen hierarchisch von links nach rechts. So wird dem Standardbenutzer zunächst der Zugriff auf alle Schlüssel und Kanäle entzogen und anschließend nur der Zugriff auf diejenigen mit dem Präfix `minimal:` gewährt. Das Gleiche gilt für die Befehlsgruppen, bei denen der Zugriff zunächst auf alle Befehle eingeräumt wird und anschließend alle administrativen und gefährlichen Befehle davon wieder ausgenommen werden.

A32 Die Vergabe von Zugriffsrechten muss nach dem Least-Privilege- und Erforderlichkeitsprinzip erfolgen.

Neo4j Neo4j verfügt in der Community-Edition nur über eine ACL mit vordefinierten Standardberechtigungen, die im Wesentlichen die folgenden Aktionen erlauben [90]:

- Zugriff, Bearbeitung und Strukturierung aller Datenbanken und deren Inhalte,
- Erstellung, Bearbeitung und Löschung aller Benutzerkonten,
- Ausführung von Prozeduren, Funktionen und administrativen Befehlen.

Der Zugriff auf diese Standardberechtigungen kann nicht angepasst werden und ist daher für alle Benutzerkonten gleichermaßen verfügbar.

Redis Die Benutzerkonten, die im Rahmen der vorherigen Umsetzung der Anforderung A31 definiert wurden, erfüllen diese Anforderung vollständig.

4.1.5 Passwortrichtlinien

A33 Verwendete Passwörter müssen hohen Sicherheitsanforderungen standhalten.

Neo4j und Redis verfügen über keine integrierte Funktionalität zur Durchsetzung einer Kennwortrichtlinie.

A34 Vordefinierte Standard-Passwörter und -Benutzerkennungen dürfen nicht verwendet werden.

Neo4j Die Standard-Benutzerkennung und das -Passwort für Neo4j sind identisch (`neo4j`). Diese können über das Bolt-Protokoll unter Verwendung der Cypher Shell wie folgt geändert werden:

```
1 # Änderung des Standard-Passworts und Änderung der
  ↪  Standard-Benutzerkennung (Cypher-Shell)
2 ALTER CURRENT USER SET PASSWORD FROM 'neo4j' TO 'examplepassword'
3 RENAME USER neo4j IF EXISTS TO exampleuser
```

Redis In Redis gibt es die Standard-Benutzerkennung `default`, für die vor der Umsetzung der Anforderung A10 kein Passwort konfiguriert ist. Da diese

allerdings nicht umbenannt werden kann, ist es notwendig, diese in der ACL-Datei /etc/redis/users.acl wie folgt zu sperren:

```
# Standard-Benutzerkennung deaktivieren
user default off resetpass resetkeys resetchannels -@all
```

A35 Passwörter und kryptografische Schlüssel dürfen nur einen einzigen Einsatzzweck aufweisen und nicht mehrfach verwendet werden.

Neo4j Neo4j speichert immer alle Passwörter als SHA-256 mit Salt in der Neo4j-Systemdatenbank. Daher kann die Konformität mit dieser Anforderung nicht überprüft werden, ohne die mit den Hash-Werten assoziierten Klartext-Passwörter zu kennen.

Redis Redis speichert unter Berücksichtigung der hier angestrebten gehärteten Konfiguration alle Passwörter als SHA-256 ohne Salt in der ACL-Datei /etc/redis/users.acl. Um die Konformität mit dieser Anforderung zu überprüfen, können daher die Hash-Werte aller Benutzerkonten wie folgt auf Duplikate untersucht werden:

```
# Suche nach Passwort-Hash-Duplikaten
cat /etc/redis/users.acl | grep -e "^user" | grep -oe
↪  "[0-9a-f]\{64\}" | uniq -c
```

Der erwartete Rückgabewert für den Fall, dass zwei Duplikate vorliegen, entspricht folgendem Muster:

```
# Muster-Ausgabe von uniq -c für Passwort-Hash-Duplikate
2    68b77a946d17400f0cca8ddd86b145015e5d01cb89c8d953bb4067adebb91fbe
```

A36 Frühere Passwörter dürfen nicht wiederverwendet werden.

Neo4j und Redis bieten keine Möglichkeit, die Wiederverwendung von früheren Passwörtern in der Community-Edition zu unterbinden.

A37 Passwörter dürfen nur mit einer sicheren Methode als Hash unter Verwendung eines Salts sowie, falls möglich, mit Peppering gespeichert werden.

Neo4j Neo4j speichert alle Passwörter immer in Hash-Form unter Verwendung von SHA-256 mit Salt in der Neo4j-Systemdatenbank. Dieses Verhalten kann nicht weiter angepasst werden.

Redis Redis unterstützt nur SHA-256 ohne Salt und Peppering für die sichere Speicherung von Passwörtern. Der Passwort-Hash-Wert muss 64 Zeichen aufweisen und in Hexadezimalzeichen in Kleinbuchstaben angegeben werden. Dies kann in der ACL-Datei /etc/redis/users.acl gemäß dem folgenden Format konfiguriert werden:

```
1 # #-Präfix für Passwörter als SHA-256-Hash
2 user example on +@all
  ↪ #68b77a946d17400f0cca8ddd86b145015e5d01cb89c8d953bb4067adebb91fbe
```

A38 Mechanismen zum Zurücksetzen von Passwörtern dürfen keine Angriffsfläche für Angreifer bieten.

Neo4j In Neo4j können Passwörter nur zurückgesetzt werden, indem zunächst die Authentifizierung deaktiviert wird, anschließend ein neues Passwort wie zuvor im Rahmen der Umsetzung der Anforderung A34 gesetzt wird und abschließend die Authentifizierung wieder aktiviert wird. Diese Vorgehensweise kann von einem Angreifer nur dann ausgenutzt werden, wenn der Zugriff auf die Konfigurationsdatei /etc/neo4j/neo4j.conf im Rahmen der Umsetzung der Anforderung A65 nicht korrekt eingeschränkt wurde.

Redis In Redis kann das Passwort auf zwei Arten zurückgesetzt werden. Zum einen ist es möglich, den Hash-Wert des Passworts in der Datei /etc/redis/users.acl zu ändern. Dieser Ansatz kann von einem Angreifer allerdings nur dann ausgenutzt werden, wenn der Zugriff auf diese Konfigurationsdatei im Rahmen der Umsetzung der Anforderung A65 nicht ordnungsgemäß eingeschränkt wurde. Alternativ dazu kann auch der Befehl `ACL SETUSER` zusammen mit `ACL SAVE` verwendet werden. Dies setzt jedoch voraus, dass eine authentifizierte Sitzung mit Zugriff auf die administrativen Befehle vorhanden ist, was bei korrekter Umsetzung der Anforderung A31 nicht zutrifft.

A39 Passwörter dürfen nicht aufgrund von zeitlichen Nutzungsbegrenzungen geändert werden.

Neo4j und Redis verfügen über keine integrierte Funktionalität zur zeitlichen Begrenzung der Nutzung bzw. zum automatischen Ablauf von Passwörtern nach einer vordefinierten Zeitspanne.

A40 Zur Erkennung von Passwortkompromittierungen müssen geeignete Schutzmaßnahmen ergriffen werden.

Neo4j Neo4j verfügt über keine integrierte Funktionalität zur Erkennung von Passwortkompromittierungen, die über eine Audit- und Protokollierungsfunktion hinausgeht.

Redis Redis verfügt ebenfalls über keine entsprechende integrierte Funktionalität. Aufgrund der Speicherung von Passwörtern als SHA-256 ohne Salt ließe sich jedoch ein Abgleich mit Rainbow-Tabellen oder der „Have I Been Pwned?“ Datenbank im Rahmen einer Eigenentwicklung eines Plug-ins/einer Software-Erweiterung verwirklichen. Dies ist jedoch nicht Teil dieser Master-Thesis.

4.1.6 Auditierung und Protokollierung

A41 Die Auditierung sowie Protokollierung muss konfiguriert und aktiv sein.

Neo4j Für Neo4j kann die Protokollierung wie folgt in der Datei /etc/neo4j/server-logs.xml aktiviert werden, indem sichergestellt wird, dass das Log-Level von `QueryLogger`, `HttpLogger` und `SecurityLogger` jeweils nicht den Wert `OFF` aufweist:

```
1  <!--Server-Protokollierung konfigurieren-->
2  <Configuration [...]>
3      [...]
4      <Loggers>
5          <Root level="INFO">
6              <AppenderRef ref="DebugLog"/>
7          </Root>
8          <Logger name="QueryLogger" level="INFO" additivity="false">
9              <AppenderRef ref="QueryLog"/>
10         </Logger>
11         <Logger name="HttpLogger" level="INFO" additivity="false">
```

```
12             <AppenderRef ref="HttpLog"/>
13         </Logger>
14         <Logger name="SecurityLogger" level="INFO" additivity="false">
15             <AppenderRef ref="SecurityLog"/>
16         </Logger>
17     </Loggers>
18 </Configuration>
```

Zusätzlich muss Selbiges in der Datei /etc/neo4j/user-logs.xml für das Log-Level von `Root` wie folgt sichergestellt werden:

```
1  <!--Benutzer-Protokollierung konfigurieren-->
2  <Configuration [...]>
3      [...]
4      <Loggers>
5          <Root level="INFO">
6              <AppenderRef ref="Neo4jLog"/>
7              <AppenderRef ref="ConsoleAppender"/>
8          </Root>
9      </Loggers>
10 </Configuration>
```

Redis Für Redis kann die Protokollierung in der Datei /etc/redis/redis.conf aktiviert werden, indem sichergestellt wird, dass das Log-Level nicht den Wert `off` aufweist und der Crash-Log aktiv ist:

```
1 # Log-Level konfigurieren (Standardwert notice)
2 loglevel verbose
3 # Crash-Log aktivieren (Standardwert no)
4 crash-log-enabled yes
```

A42 Sind mehrere Audit- und Protokollierungsfunktionen verfügbar, ist das sicherste Verfahren zu verwenden.

Neo4j Neo4j unterstützt nur die Protokollierung mittels Log4j Version 2 über entsprechende Protokolldateien. Die Protokollierung über wird in diesem Zusammenhang nicht direkt unterstützt, kann aber im Rahmen einer Eigenentwicklung implementiert werden. Dies ist jedoch nicht Teil dieser Master-Thesis.

Redis Für Redis kann die Protokollierung mittels Syslog bzw. systemd/Jour-

nal in der Datei /etc/redis/redis.conf wie folgt aktiviert werden:

```
1 # Syslog konfigurieren (Standardwert <auskommentiert>)
2 syslog-enabled yes
3 syslog-ident redis
4 syslog-facility local0
```

A43 Für Audit-Protokolle muss ausreichend Speicherplatz bereitgestellt werden.

Für Neo4j und Redis kann dies umgesetzt werden, indem überprüft wird, dass /var/log eine separate Partition mit mindestens 50 GB Gesamtspeicherkapazität ist.

A44 Audit-Protokolle müssen in ein separates Log-Management-System ausgelagert werden.

Für Neo4j und Redis ist kein integrierter Mechanismus zur Weiterleitung von Protokolldateien an ein separates Log-Management-System vorhanden. Durch die Verwendung von Syslog und/oder Protokolldateien sind jedoch bereits alle notwendigen Voraussetzungen gegeben, um diese Funktionalität mittels geeigneter Dienste realisieren zu können.

A45 Um den Verlust von Audit-Protokollen zu verhindern, müssen Warnungen gesendet werden, wenn der Speicherplatz knapp wird oder die Protokollierung fehlschlägt.

Neo4j und Redis unterstützen in der Community-Edition keine Konfiguration von Triggern/Warnungen zur Benachrichtigung über sich abzeichnende Speicherplatzprobleme. Stattdessen ist es jedoch möglich, ergänzend zur Überprüfung der Gesamtspeicherkapazität von /var/log im Rahmen der Umsetzung der Anforderung A43 zu überprüfen, ob mindestens 25 % des reservierten Speichers nicht in Anspruch genommen werden.

A46 Audit-Protokolle sind geordnet nach ihrem Alter zu überschreiben, wenn der Speicherplatz für neue Einträge erschöpft ist.

Neo4j und Redis unterstützen in der Community-Edition kein kontrolliertes Überschreiben von Protokollinformationen nach Alter, sobald der Speicherplatz für neue Einträge erschöpft ist.

A47 Audit-Protokolle müssen alle Ereignisse und Aktivitäten erfassen (maximale Verbosität).

Neo4j Für Neo4j wurde dies bereits in Teilen durch die Umsetzung der Anforderung A41 implementiert. Ein Log-Level `VERBOSE` ist dort für die Konfiguration von Log4j nicht verfügbar, weshalb der Wert `INFO` einen guten Kompromiss darstellt. Zusätzlich müssen noch die folgenden Konfigurationsoptionen in der Datei /etc/neo4j/neo4j.conf angepasst werden:

```
1  # Aktivierung des HTTP-Protokolls (Standardwert false)
2  dbms.logs.http.enabled=true
3  # Aktivierung des Query-Protokolls (Standardwert VERBOSE)
4  db.logs.query.enabled=VERBOSE
5  # Aktivierung des Garbage-Collection-Protokolls (Standardwert false)
6  server.logs.gc.enabled=true
7  # Konfiguration der Garbage-Collection Optionen
8  server.logs.gc.options=-Xlog:gc*,safepoint,age*=trace
9  # Konfiguration der max. Aufbewahrung von Dateien (Standardwert 5)
10 server.logs.gc.rotation.keep_number=50
11 # Konfiguration der maximalen Protokollgröße (Standardwert 20m)
12 server.logs.gc.rotation.size=50m
```

Redis Für Redis wurde dies bereits durch die Umsetzung der Anforderung A41 implementiert, indem der Konfigurationsparameter `loglevel` auf den Wert `verbose` abgeändert wurde.

A48 Audit-Protokolle müssen einem vordefinierten Format entsprechen, das ihre Analyse erleichtert.

Neo4j Für Neo4j kann das Protokollierungsformat wie folgt in der Datei /etc/neo4j/server-logs.xml konfiguriert werden:

```
1  <!--Server-Protokollierungsformat konfigurieren-->
2  <Configuration [...]>
3      <Appenders>
4          <RollingRandomAccessFile
5          name="DebugLog"
           ↪  fileName="${config:server.directories.logs}/debug.log"
6          filePattern="$${config:server.directories.logs}/debug.log.%02
           ↪  i">
7          <Neo4jDebugLogLayout pattern="%d{yyyy-MM-dd
           ↪  HH:mm:ss.SSSZ}{GMT+0} %-5p [%c{1.}] %m%n"/>
```

```
8          <Policies>
9          <SizeBasedTriggeringPolicy size="50 MB"/>
10         </Policies>
11         <DefaultRolloverStrategy fileIndex="min" max="50"/>
12         </RollingRandomAccessFile>
13         <RollingRandomAccessFile
14         name="HttpLog"
           ↪  fileName="${config:server.directories.logs}/http.log"
15         filePattern="$${config:server.directories.logs}/http.log.%02i
           ↪  ">
16         <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSSZ}{GMT+0}
           ↪  %-5p %m%n"/>
17         <Policies>
18         <SizeBasedTriggeringPolicy size="50 MB"/>
19         </Policies>
20         <DefaultRolloverStrategy fileIndex="min" max="50"/>
21         </RollingRandomAccessFile>
22         <RollingRandomAccessFile
23         name="QueryLog"
           ↪  fileName="${config:server.directories.logs}/query.log"
24         filePattern="$${config:server.directories.logs}/query.log.%02
           ↪  i">
25         <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSSZ}{GMT+0}
           ↪  %-5p %m%n"/>
26         <Policies>
27         <SizeBasedTriggeringPolicy size="50 MB"/>
28         </Policies>
29         <DefaultRolloverStrategy fileIndex="min" max="50"/>
30         </RollingRandomAccessFile>
31         <RollingRandomAccessFile
32         name="SecurityLog"
           ↪  fileName="${config:server.directories.logs}/security.log"
33         filePattern="$${config:server.directories.logs}/security.log.
           ↪  %02i">
34         <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSSZ}{GMT+0}
           ↪  %-5p %m%n"/>
35         <Policies>
36         <SizeBasedTriggeringPolicy size="50 MB"/>
37         </Policies>
38         <DefaultRolloverStrategy fileIndex="min" max="50"/>
39         </RollingRandomAccessFile>
40     </Appenders>
41     [...]
42 </Configuration>
```

Zusätzlich muss Selbiges in der Datei /etc/neo4j/user-logs.xml wie folgt umgesetzt werden:

```
1  <!--Benutzer-Protokollierungsformat konfigurieren-->
2  <Configuration [...]>
3      <Appenders>
4          <RollingRandomAccessFile
5          name="Neo4jLog"
           ↪  fileName="${config:server.directories.logs}/neo4j.log"
6          filePattern="$${config:server.directories.logs}/neo4j.log.%02
           ↪  i">
7          <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSSZ}{GMT+0}
           ↪  %-5p %m%n"/>
8          <Policies>
9          <SizeBasedTriggeringPolicy size="50 MB"/>
10         </Policies>
11         <DefaultRolloverStrategy fileIndex="min" max="50"/>
12         </RollingRandomAccessFile>
13         <Console name="ConsoleAppender" target="SYSTEM_OUT">
14         <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSSZ}{GMT+0}
           ↪  %-5p %m%n"/>
15         </Console>
16     </Appenders>
17     [...]
18 </Configuration>
```

Redis Für Redis wurde dies bereits durch die Umsetzung der Anforderung A42 konfiguriert, da durch die Aktivierung von Syslog bzw. systemd/Journal ein standardisiertes Protokollformat automatisch angewendet wird.

A49 Audit-Protokolle müssen in einem Verzeichnis mit leicht zuzuordnenden Dateinamen gespeichert werden.

Neo4j Für Neo4j wurde dies bereits durch die Umsetzung der Anforderung A48 konfiguriert, indem der relative Pfad mit der Konfigurationsoption `fileName` und das Nummerierungsschema mit `filePattern` angegeben wurde. Zusätzlich muss noch die folgende Option in der Datei /etc/neo4j/neo4j.conf für den absoluten Speicherpfad angepasst werden:

```
1  # Absoluter Log-Speicherpfad konfigurieren (Standardwert
   ↪  /var/log/neo4j)
2  server.directories.logs=/var/log/neo4j
```

Redis Für Redis kann dies in der Datei /etc/redis/redis.conf wie folgt sichergestellt werden:

```
# Log-Speicherpfad konfigurieren (Standardwert
↪ /var/log/redis/redis-server.log)
logfile /var/log/redis/redis-server.log
```

A50 Der Zugriff auf die Konfiguration der Auditierung und Protokollierung muss begrenzt werden.

Neo4j In Neo4j ist es nicht möglich, direkt aus der Anwendung heraus auf die Konfiguration der Auditierung und Protokollierung zuzugreifen und diese zu bearbeiten.

Redis In Redis kann der Befehl `CONFIG SET [Parameterwert …]` zusammen mit `CONFIG REWRITE` verwendet werden, um alle Konfigurationsoptionen während der Laufzeit anzupassen. Dies setzt jedoch voraus, dass eine authentifizierte Sitzung mit Zugriff auf die administrativen Befehle vorhanden ist, was bei korrekter Umsetzung der Anforderung A31 nicht zutrifft.

A51 Der Zugriff auf die Inhalte der Auditierung und Protokollierung muss begrenzt werden.

Neo4j und Redis bieten in der Community-Edition keine Möglichkeit, direkt aus der Anwendung heraus auf die Protokollierungsdaten zuzugreifen und diese anzuzeigen.

4.1.7 Monitoring

A52 Alle kritischen Parameter, Ereignisse und Betriebszustände müssen überwacht werden.

Als Voraussetzung für die Umsetzung der Anforderungen A42 und A43 muss der Linux-Audit-Daemon als Basis für die Protokollierung sicherheitsrelevanter Informationen installiert werden:

```
# Linux Audit Daemon installieren
apt update; apt install auditd
```

Anschließend müssen in der Datei /etc/audit/auditd.conf die nachstehenden

Konfigurationsoptionen wie folgt angepasst werden [91]:

```
# Linux Audit Daemon konfigurieren
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_error_action = SUSPEND
disk_full_action = SUSPEND
flush = INCREMENTAL_ASYNC
log_file = /var/log/audit/audit.log
log_format = raw
max_log_file_action = keep_logs
space_left = 75
space_left_action = SYSLOG
```

Eine Möglichkeit, kritische Parameter und Betriebszustände etwa per SNMP zu überwachen, bietet die Community-Edition von Neo4j und Redis jedoch nicht, kann aber im Rahmen einer Eigenentwicklung implementiert werden. Dies ist jedoch nicht Teil dieser Master-Thesis.

4.1.8 Fingerprinting

A53 Der Debug-Modus muss deaktiviert werden.

Neo4j Für Neo4j kann der Debug-Modus in der Datei /etc/neo4j/neo4j.conf deaktiviert werden, indem die folgende Zeile auskommentiert oder entfernt wird:

```
# Debug-Modus deaktivieren (Standardwert <auskommentiert>)
#dbms.jvm.additional=-agentlib:jdwp=transport=dt_socket,server=y,susp
↪ end=n,address=*:5005
```

Die aufgeführten Argumente für das Java Debug Wire Protocol (JDWP) haben die folgende Funktion [92]:

transport=dt_socket Name der Schnittstelle, die für die Verbindung zur Debugger-Anwendung verwendet werden soll.

server=y Legt fest, ob die Java Virtual Machine auf die Verbindung einer Debugger-Anwendung warten soll (`y`) oder selbst eine Verbindung zu dieser aufbauen soll (`n`).

suspend=n Legt fest, ob die Java Virtual Machine mit der Ausführung der Anwendung beginnen soll (`n`) oder ob dafür eine aktive Verbindung zur Debugger-Anwendung bestehen muss (`y`).

address=*:5005 Angabe des Eingangsports für die Verbindung durch die Debugger-Anwendung.

Redis Für Redis kann der Debug-Modus nicht deaktiviert werden, jedoch der Zugriff auf diesen beschränkt werden. Dazu muss sichergestellt werden, dass für alle nicht-administrativen Benutzer das Argument `-DEBUG` oder `-@dangerous` in der ACL-Datei /etc/redis/users.acl wie folgt enthalten ist:

```
1 # Debug-Modus deaktivieren
2 user example on +@all -DEBUG
  ↪ #68b77a946d17400f0cca8ddd86b145015e5d01cb89c8d953bb4067adebb91fbe
3 user example on +@all -@dangerous
  ↪ #68b77a946d17400f0cca8ddd86b145015e5d01cb89c8d953bb4067adebb91fbe
```

A54 Die Ausgabe von Fehlermeldungen darf nicht zur Durchführung von Angriffen interpretiert werden können.

Neo4j und Redis geben ohne eine authentifizierte Sitzung im Fehlerfall keine Informationen zurück, die zur Durchführung von Angriffen interpretiert werden können. Bei erfolgter Authentifizierung beschränken sich die Fehlermeldungen auf die notwendigsten Informationen. Dieses Verhalten kann in den Community-Editionen nicht weiter angepasst werden.

A55 Die Verbindung zum Datenbanksystem darf keine Rückschlüsse auf die Version zulassen.

Neo4j Neo4j gibt beim Verbindungsaufbau zur HTTPS- und Bolt-Schnittstelle Informationen zurück, die zur Identifizierung des Datenbanksystems verwendet werden können, jedoch nicht zur genauen Bestimmung der Version im nicht authentifizierten Zustand. Dieses Verhalten kann nicht weiter angepasst werden.

Redis Redis liefert bei einem Verbindungsaufbau keine Informationen zurück, bis der Initiator der Verbindung selbst Daten an Redis sendet. Der erwartete Rückgabewert entspricht:

```
# Redis Standard-Fehlermeldung
-ERR unknown command 'example', with args beginning with:
```

Dies könnte zur Identifizierung des Datenbanksystems genutzt werden, da die Struktur des Strings für Redis spezifisch ist, jedoch nicht zur genauen Bestimmung der Version im nicht authentifizierten Zustand. Dieses Verhalten kann nicht weiter angepasst werden.

4.1.9 Verschlüsselung

A56 Die Kommunikation über Schnittstellen muss verschlüsselt erfolgen.

Neo4j Für Neo4j kann die Verschlüsselung der Kommunikation in der Datei /etc/neo4j/neo4j.conf für alle Schnittstellen wie folgt sichergestellt werden:

```
# HTTP deaktivieren (Standardwert true)
server.http.enabled=false
#server.http.listen_address=localhost:8574
#server.http.advertised_address=localhost:8574

# HTTPS aktivieren (Standardwert false)
server.https.enabled=true
# Bolt aktivieren (Standardwert true)
server.bolt.enabled=true
# Bolt TLS-Level konfigurieren (Standardwert DISABLED)
server.bolt.tls_level=REQUIRED
```

Redis Für Redis kann die Verschlüsselung der Kommunikation in der Datei /etc/redis/redis.conf für alle Schnittstellen wie folgt sichergestellt werden:

```
# Port deaktivieren (Standardwert 6379)
port 0
# TLS-Port aktivieren (Standardwert 6379)
tls-port 7479
```

A57 Die Kommunikation aller Teilnehmer in einem Cluster muss verschlüsselt erfolgen.

Neo4j und Redis ermöglichen in der Community-Edition keinen Betrieb von Clustern.

A58 Die Verschlüsselung muss mit sicheren kryptografischen Protokollen betrieben werden.

Neo4j Für Neo4j können die kryptografischen Protokolle in der Datei /etc/-neo4j/neo4j.conf wie folgt festgelegt werden:

```
1 # Bolt TLS-Protokolle konfigurieren (Standardwert TLSv1.2)
2 dbms.ssl.policy.bolt.tls_versions=TLSv1.3,TLSv1.2
3 # HTTPS TLS-Protokolle konfigurieren (Standardwert TLSv1.2)
4 dbms.ssl.policy.https.tls_versions=TLSv1.3,TLSv1.2
```

Redis Für Redis können die kryptografischen Protokolle in der Datei /etc/redis/redis.conf wie folgt festgelegt werden:

```
1 # TLS-Protokolle konfigurieren (Standardwert "TLSv1.2 TLSv1.3")
2 tls-protocols "TLSv1.3 TLSv1.2"
```

A59 Das für den Schlüsselaustausch verwendete Verfahren muss sicher sein.

Für Neo4j und Redis wird durch die Umsetzung der Anforderungen A58 und A60 die Verwendung von besonders sicheren kryptografischen Algorithmen sowohl direkt als auch indirekt erzwungen. Diese werden nachfolgend in Anforderung A60 genau festgelegt und unterstützen PFS. Daher müssen Diffie-Hellman (DH)-Parameter für deren DH-Gruppen generiert werden, da andernfalls die PFS-Algorithmen automatisch deaktiviert werden [93].

Neo4j Für Neo4j kann die Verwendung einer statischen DH-Parameter-Datei in der Datei /etc/neo4j/neo4j.conf wie folgt aktiviert werden:

```
1 # DH-Params aktivieren (Standardwert <einkommentiert>)
2 server.jvm.additional=-Djdk.tls.ephemeralDHKeySize=2048
```

Die Verwendung einer DH-Parameterdatei mit mehr als 2048 Bit wird von

Java und somit auch von Neo4j bislang nicht unterstützt.

Redis Für Redis kann die Verwendung einer statischen DH-Parameter-Datei in der Datei /etc/redis/redis.conf wie folgt aktiviert werden:

```
1 # DH-Params aktivieren (Standardwert <auskommentiert>)
2 tls-dh-params-file redis-4096.dh
```

A60 Die Verschlüsselung muss mit sicheren kryptografischen Algorithmen betrieben werden.

Neo4j Für Neo4j können die kryptografischen Algorithmen in der Datei /etc/neo4j/neo4j.conf wie folgt festgelegt werden:

```
1 # Bolt TLSv1.3- und TLSv1.2-Algorithmen konfigurieren (Standardwert
  ↪ <leer>)
2 dbms.ssl.policy.bolt.ciphers=TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_S
  ↪ HA256,TLS_AES_128_CCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA
  ↪ 384,TLS_ECDHE_ECDSA_WITH_AES_256_CCM,TLS_ECDHE_RSA_WITH_AES_256_G
  ↪ CM_SHA384,TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AE
  ↪ S_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_256_CCM,TLS_ECDHE_ECDSA_WIT
  ↪ H_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CCM,TLS_ECDHE_R
  ↪ SA_WITH_AES_128_GCM_SHA256,TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,TL
  ↪ S_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_128_CCM
3 # HTTPS TLSv1.3- und TLSv1.2-Algorithmen konfigurieren (Standardwert
  ↪ <leer>)
4 dbms.ssl.policy.https.ciphers=TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_
  ↪ SHA256,TLS_AES_128_CCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SH
  ↪ A384,TLS_ECDHE_ECDSA_WITH_AES_256_CCM,TLS_ECDHE_RSA_WITH_AES_256_
  ↪ GCM_SHA384,TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_A
  ↪ ES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_256_CCM,TLS_ECDHE_ECDSA_WI
  ↪ TH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CCM,TLS_ECDHE_
  ↪ RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,T
  ↪ LS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_128_CCM
```

Redis Für Redis können die kryptografischen Algorithmen in der Datei /etc/redis/redis.conf wie folgt festgelegt werden:

```
1 # TLSv1.3-Algorithmen konfigurieren (Standardwert
  ↪ TLS_CHACHA20_POLY1305_SHA256)
2 tls-ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:
  ↪ TLS_AES_128_GCM_SHA256
```

```
3 # TLSv1.2-Algorithmen konfigurieren (Standardwert DEFAULT:!MEDIUM)
4 tls-ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384
  ↪ :DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RS
  ↪ A-CHACHA20-POLY1305:DHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-
  ↪ GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256
5 # Server-Präferenz aktivieren (Standardwert yes)
6 tls-prefer-server-ciphers yes
```

A61 Die kryptografischen Algorithmen müssen eine hohe Schlüssellänge aufweisen.

Im Rahmen der Umsetzung der Anforderung A60 wurden bereits alle technischen Maßnahmen zur Erfüllung dieser Anforderung für Neo4j und Redis umgesetzt.

A62 Selbstsignierte Zertifikate dürfen für eine verschlüsselte Kommunikation nicht verwendet und akzeptiert werden.

Neo4j Für Neo4j können die anzuwendenden offiziellen Zertifikate in der Datei /etc/neo4j/neo4j.conf wie folgt festgelegt werden:

```
1 # Bolt TLS-Zertifikate konfigurieren (Standardwert <auskommentiert>)
2 dbms.ssl.policy.bolt.enabled=true
3 dbms.ssl.policy.bolt.base_directory=certificates/bolt
4 dbms.ssl.policy.bolt.private_key=private.key
5 dbms.ssl.policy.bolt.public_certificate=public.crt
6 dbms.ssl.policy.bolt.trust_all=false
7
8 # HTTPS TLS-Zertifikate konfigurieren (Standardwert <auskommentiert>)
9 dbms.ssl.policy.https.enabled=true
10 dbms.ssl.policy.https.base_directory=certificates/https
11 dbms.ssl.policy.https.private_key=private.key
12 dbms.ssl.policy.https.public_certificate=public.crt
13 dbms.ssl.policy.https.trust_all=false
```

Redis Für Redis können die anzuwendenden offiziellen Zertifikate in der Datei /etc/redis/redis.conf wie folgt festgelegt werden:

```
1 # TLS-Zertifikate konfigurieren (Standardwert <auskommentiert>)
2 tls-cert-file redis.crt
3 tls-key-file redis.key
```

```
4 tls-ca-cert-file ca.crt
```

A63 Für die Erstellung von Anmeldeinformationen und Zertifikaten müssen sichere Schlüsselgeneratoren verwendet werden.

Neo4j Neo4j verfügt nicht über integrierte Funktionalitäten, die einen direkten Zugriff auf einen Schlüsselgenerator erfordern. Stattdessen müssen alle Anmeldeinformationen oder Zertifikate vom Benutzer selbst erstellt und anschließend hinterlegt werden.

Redis Redis nutzt für den Befehl `ACL GENPASS` den Schlüsselgenerator /dev/urandom. Dieser kann nicht individuell angepasst oder anderweitig konfiguriert werden.

A64 Die Anwendungsdaten müssen verschlüsselt werden.

Neo4j und Redis verfügen in der Community-Edition nicht über eine Verschlüsselung von ruhenden Anwendungsdaten. Bei Bedarf muss dies seitens des Betriebssystems auf Dateisystemebene oder durch die angebundene Anwendung erfolgen.

4.1.10 Verzeichnis- und Dateiberechtigungen

A65 Die Zugriffsrechte auf zur Datenbankanwendung gehörende Verzeichnisse, Dateien und Anwendungen müssen restriktiv vergeben werden.

Neo4j Für Neo4j können in der Datei /etc/neo4j/neo4j.conf die absoluten Speicherpfade für die Datenbankinhalte, Plug-ins und Anwendungsbibliotheken wie folgt konfiguriert werden:

```
1 # Pfad der Datenbankinhalte konfigurieren (Standardwert
  ↪ /var/lib/neo4j/data)
2 server.directories.data=/var/lib/neo4j/data
3 # Pfad der Plug-ins konfigurieren (Standardwert
  ↪ /var/lib/neo4j/plugins)
4 server.directories.plugins=/var/lib/neo4j/plugins
5 # Pfad der Anwendungsbibliotheken konfigurieren (Standardwert
  ↪ /usr/share/neo4j/lib)
6 server.directories.lib=/usr/share/neo4j/lib
```

Für einen sicheren Betrieb muss sichergestellt werden, dass

- alle Verzeichnisse und Dateien unter /var/lib/neo4j/, /usr/share/neo4j/ und /etc/neo4j/ (rekursiv) dem Systembenutzer neo4j und der -gruppe neo4j oder adm gehören.
- alle Verzeichnisse unter /var/lib/neo4j/, /usr/share/neo4j/ und /etc/-neo4j/ (rekursiv) nur Schreibrechte für den Eigentümer aufweisen, d. h. in oktaler Notation maximal 0750 oder weniger.
- alle Dateien unter /var/lib/neo4j/ und /etc/neo4j/ (rekursiv) nur Schreibrechte für den Eigentümer und grundsätzlich keine Ausführungsrechte aufweisen, d. h. in oktaler Notation maximal 0640 oder weniger.
- alle Dateien unter /usr/share/neo4j/ (rekursiv) nur Schreibrechte für den Eigentümer und Ausführungsrechte nur für den Eigentümer und die Gruppe aufweisen, d. h. in oktaler Notation maximal 0750 oder weniger.

Redis Für Redis kann in der Datei /etc/redis/redis.conf der absolute Speicherpfad für die Datenbankinhalte wie folgt konfiguriert werden:

```
1 # Pfad der Datenbankinhalte konfigurieren (Standardwert
  ↪ /var/lib/redis)
2 dir /var/lib/redis
```

Für einen sicheren Betrieb muss sichergestellt werden, dass

- alle Verzeichnisse und Dateien unter /var/lib/redis/ und /etc/redis/ (rekursiv) dem Systembenutzer redis und der -gruppe redis gehören.
- alle Verzeichnisse unter /var/lib/redis/ und /etc/redis/ (rekursiv) nur Schreibrechte für den Eigentümer aufweisen, d. h. in oktaler Notation maximal 0750 oder weniger.
- alle Dateien unter /var/lib/redis/ und /etc/redis/ (rekursiv) nur Schreibrechte für den Eigentümer und grundsätzlich keine Ausführungsrechte aufweisen, d. h. in oktaler Notation maximal 0640 oder weniger.

A66 Die Zugriffsrechte auf Protokollierungsdaten müssen restriktiv vergeben werden.

Neo4j Für Neo4j wurde im Rahmen der Umsetzung der Anforderung A49 bereits der absolute Pfad der Protokollierungsdaten konfiguriert. Für einen

sicheren Betrieb muss sichergestellt werden, dass

- alle Verzeichnisse und Dateien unter /var/log/neo4j/ (rekursiv) dem Systembenutzer neo4j und der -gruppe neo4j oder adm gehören.
- alle Verzeichnisse unter /var/log/neo4j/ (rekursiv) nur Schreibrechte für den Eigentümer aufweisen, d. h. in oktaler Notation maximal 0750 oder weniger.
- alle Dateien unter /var/log/neo4j/ (rekursiv) nur Schreibrechte für den Eigentümer und grundsätzlich keine Ausführungsrechte aufweisen, d. h. in oktaler Notation maximal 0640 oder weniger.

Redis Für Redis wurde im Rahmen der Umsetzung der Anforderung A49 bereits der absolute Pfad der Protokollierungsdaten konfiguriert. Für einen sicheren Betrieb muss sichergestellt werden, dass

- alle Verzeichnisse und Dateien unter /var/log/redis/ (rekursiv) dem Systembenutzer redis und der -gruppe redis oder adm gehören.
- alle Verzeichnisse unter /var/log/redis/ (rekursiv) nur Schreibrechte für den Eigentümer aufweisen, d. h. in oktaler Notation maximal 0750 oder weniger.
- alle Dateien unter /var/log/redis/ (rekursiv) nur Schreibrechte für den Eigentümer und grundsätzlich keine Ausführungsrechte aufweisen, d. h. in oktaler Notation maximal 0640 oder weniger.

A67 Die Zugriffsrechte auf kryptografische Schlüssel müssen restriktiv vergeben werden.

Für Neo4j und Redis wurde diese Anforderung bereits indirekt durch die Umsetzung der Anforderung A65 umgesetzt, da zuvor alle kryptografischen Schlüssel und Passwortdateien unter /etc/neo4j/ bzw. /etc/redis/ abgelegt wurden. Mit der Erfüllung der referenzierten Anforderung wird damit gleichzeitig auch diese Anforderung erfüllt.

4.1.11 Sicherer Betrieb der Datenbankanwendung

A68 Datenbankspezifische Schutzmechanismen müssen konfiguriert und aktiviert werden.

Für Neo4j und Redis stehen in der Community-Edition keine zusätzlichen

Maßnahmen zur Verfügung, die über die bereits getroffenen und noch zu treffenden Maßnahmen im Rahmen der hier angestrebten gehärteten Konfiguration hinausgehen.

A69 Funktionen, die die Ausführung von dynamischem Code verhindern, müssen aktiviert werden.

Neo4j In Neo4j kann die Ausführung von Prozeduren auf eine Liste von erlaubten Prozeduren beschränkt werden. Diese sollten in der Datei /etc/neo4j/-neo4j.conf auf die standardmäßig vorhandenen Awesome Procedures On Cypher (APOC)- und Graph Data Science (GDS)-Algorithmen wie folgt limitiert werden:

```
1 # Liste der standardmäßig zu ladenden Prozeduren konfigurieren
  ↪ (Standardwert *)
2 dbms.security.procedures.allowlist=apoc.*,gds.*
```

Außerdem ist es möglich, mit dem Befehl `LOAD CSV` Daten aus einer CSV-Datei zu importieren. Dies könnte es einem Angreifer ermöglichen, Schadcode zu importieren oder bestehende Daten zu überschreiben, sodass die Funktion deaktiviert werden muss. In der Community-Edition von Neo4j ist dies jedoch nicht gänzlich möglich. Stattdessen kann hingegen der Zugriff auf diese Funktion vollständig unterbunden werden, was in Teilen bereits durch die Umsetzung der Anforderung A65 durch die rekursive Begrenzung der Zugriffsrechte auf den Import-Pfad umgesetzt wurde. Schließlich muss in der Datei /etc/neo4j/-neo4j.conf der Import-Pfad angegeben und der CSV-Import über eine externe URL wie folgt deaktiviert werden:

```
1 # CSV-Import-Pfad konfigurieren (Standardwert /var/lib/neo4j/import)
2 server.directories.import=/var/lib/neo4j/import
3 # URL-CSV-Import deaktivieren (Standardwert true)
4 dbms.security.allow_csv_import_from_file_urls=false
```

Redis Für Redis stehen in der Community-Edition keine entsprechenden Maßnahmen zur Verfügung.

A70 Die Ausführung von Datenbank-Skripten muss deaktiviert werden, oder die Skripte müssen umfassend auf Schwachstellen geprüft werden.

Neo4j In Neo4j kann die Ausführung von benutzerdefinierten Prozeduren in der Datei /etc/neo4j/neo4j.conf deaktiviert werden, indem wie folgt sichergestellt wird, dass die entsprechende Konfigurationsoption keine Werte enthält:

```
1 # Liste von benutzerdefinierten Prozeduren (Standardwert <leer>)
2 dbms.security.procedures.unrestricted=
```

Redis Für Redis kann die Ausführung von Datenbank-Skripten deaktiviert werden, indem für alle nicht-administrativen Benutzer sichergestellt wird, dass das Argument `-@scripting` in der ACL-Datei /etc/redis/users.acl wie folgt enthalten ist:

```
1 # Ausführung von Datenbank-Skripten deaktivieren
2 user example on +@all -@scripting
  ↪ #68b77a946d17400f0cca8ddd86b145015e5d01cb89c8d953bb4067adebb91fbe
```

A71 Die verfügbaren Systemressourcen müssen für den Datenbankbetrieb optimiert werden.

Für Neo4j und Redis kann die Nutzung von Systemressourcen in der Community-Edition nur nach unten hin begrenzt werden. Es ist jedoch nicht möglich, diese im Hinblick auf eine Single-Tenancy-Architektur nach oben zu erweitern.

A72 Die Datenbank muss erfolgreich initialisiert werden.

Für Neo4j und Redis können in der Community-Edition keine Statusinformationen über den aktuellen Betriebszustand des Datenbanksystems abgefragt oder überwacht werden, die über die bereits implementierten Anforderungen hinausgehen.

A73 Die Datenbank muss in einen stabilen Zustand übergehen, sollte die Initialisierung fehlschlagen.

Neo4j Für Neo4j stehen in der Community-Edition keine Methoden oder Betriebsarten zur Verfügung, die bei einer Systemstörung einen Übergang in einen konsistenten Zustand sicherstellen.

Redis In Redis kann der „Append-only“ Modus verwendet werden, um sicherzustellen, dass die Datenbank im Falle eines Dienstausfalls immer in einen stabilen Zustand zurückversetzt werden kann. Dies kann in der Datei /etc/redis/redis.conf wie folgt aktiviert werden:

```
# "Append-only" Modus aktivieren (Standardwert no)
appendonly yes
# "Append-only" Intervall konfigurieren (Standardwert always)
appendfsync everysec
```

A74 Die Datenbankanwendung muss unter eigenem Benutzer und eigener Gruppe ausgeführt werden.

Neo4j Neo4j wird standardmäßig unter einem eigenen Systembenutzer und einer eigenen -gruppe mit der Bezeichnung `neo4j` ausgeführt. Dies kann in der zugehörigen systemd-Dienstdatei /usr/lib/systemd/system/neo4j.service überprüft werden, indem die Argumente `User=` und `Group=` nach dem folgenden Muster überprüft werden:

```
# systemd-Dienstdatei konfigurieren
[Unit]
Description=Neo4j Graph Database
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/usr/share/neo4j/bin/neo4j console
Restart=on-abnormal
User=neo4j
Group=neo4j

[...]
```

Redis Redis wird standardmäßig unter einem eigenen Systembenutzer und einer eigenen -gruppe mit der Bezeichnung `redis` ausgeführt. Dies kann in der zugehörigen systemd-Dienstdatei /usr/lib/systemd/system/redis-server.service überprüft werden, indem die Argumente `User=` und `Group=` nach dem folgenden Muster überprüft werden:

```
# systemd-Dienstdatei konfigurieren
[Unit]
Description=Advanced key-value store
After=network.target
Documentation=http://redis.io/documentation, man:redis-server(1)

[Service]
Type=notify
ExecStart=/usr/bin/redis-server /etc/redis/redis.conf --supervised
↪ systemd --daemonize no
PIDFile=/run/redis/redis-server.pid
TimeoutStopSec=0
Restart=always
User=redis
Group=redis

[...]
```

A75 Die Datenbankanwendung muss mit möglichst geringen Berechtigungen ausgeführt werden.

Die im Rahmen der Umsetzung der Anforderung A74 identifizierten/konfigurierten Systembenutzer und -gruppen müssen auf ihre Eigenschaften untersucht werden. Bei beiden muss sichergestellt werden, dass diese nicht dem Systembenutzer und der -gruppe root entsprechen. Außerdem muss überprüft werden, dass der Systembenutzer weder der Gruppe root noch der Gruppe sudo/wheel angehört.

A76 Die Datenbankanwendung darf nicht an 0.0.0.0 bzw. [::] gebunden werden.

Neo4j Für Neo4j kann die Standard-IP-Bindung in der Datei /etc/neo4j/-neo4j.conf für die aufgeführten Schnittstellen wie folgt angepasst werden, wobei der Doppelpunkt als Trennzeichen zwischen der IP-Adresse und der Portnummer fungiert:

```
# Anwendungsweite Standard-IP-Bindung konfigurieren (Standardwert
↪ localhost)
server.default_listen_address=localhost
server.default_advertised_address=localhost
# HTTPS konfigurieren (Standardwert :7473)
server.https.listen_address=localhost:7473
```

```
6 server.https.advertised_address=localhost:7473
7 # Bolt konfigurieren (Standardwert :7687)
8 server.bolt.listen_address=localhost:7687
9 server.bolt.advertised_address=localhost:7687
```

Redis Für Redis kann die Standard-IP-Bindung in der Datei /etc/redis/redis.conf wie folgt angepasst werden, wobei das Präfix „-“ ein Fehlschlagen des Programmstarts verhindert, wenn systemseitig keine IPv6-Unterstützung vorhanden ist:

```
1 # IP-Bindung konfigurieren (Standardwert 0.0.0.0)
2 bind 127.0.0.1 -::1
```

Darüber hinaus kann der geschützte Modus in der Datei /etc/redis/redis.conf aktiviert werden, um die IP-Bindung an die Standardroute zu unterbinden, wenn die Authentifizierung deaktiviert ist:

```
1 # Geschützten Modus aktivieren (Standardwert yes)
2 protected-mode yes
```

A77 Die Datenbankanwendung darf nicht an den Standard-Port gebunden werden.

Für die Änderung der Standard-Ports wurde ein Offset von 1 100 gegenüber dem Standardwert gewählt, um Überschneidungen entgegenzuwirken.

Neo4j Für Neo4j kann der Standard-Port in der Datei /etc/neo4j/neo4j.conf für die aufgeführten Schnittstellen wie folgt angepasst werden:

```
1 # HTTPS konfigurieren (Standardwert :7473)
2 server.https.listen_address=localhost:8573
3 server.https.advertised_address=localhost:8573
4 # Bolt konfigurieren (Standardwert :7687)
5 server.bolt.listen_address=localhost:8787
6 server.bolt.advertised_address=localhost:8787
```

Redis Für Redis kann der Standard-Port in der Datei /etc/redis/redis.conf wie folgt angepasst werden:

```
# Port deaktivieren (Standardwert 6379)
port 0
# TLS-Port abändern (Standardwert 6379)
tls-port 7479
```

A78 Die Management-Schnittstelle muss sich in einem dedizierten Netzwerksegment befinden und der Zugriff begrenzt werden.

Neo4j Zuvor wurde im Rahmen der Umsetzung der Anforderung A76 die HTTPS-Management-Schnittstelle an den lokalen Host gebunden. Neo4j unterstützt lediglich die Bindung an eine einzige Netzwerkschnittstelle (abgesehen von der Standardroute). Im Bedarfsfall kann die IP-Bindung an eine andere private IP-Adresse in gleicher Weise erfolgen, sofern die Begrenzung des Zugriffs gewahrt bleibt.

Redis Redis verfügt in der Community-Edition nicht über eine separate Management-Schnittstelle. Sowohl der Zugriff auf die Konfiguration als auch auf den Datenbankinhalt erfolgt über die selbe IP-Port-Bindung.

A79 Die Systemd-Dienstdateien müssen aktiviert werden.

Neo4j Für Neo4j kann dies mit dem folgenden Befehl umgesetzt werden:

```
# Neo4j systemd-Dienstdatei aktivieren
systemctl enable neo4j.service
```

Redis Für Redis kann dies mit dem folgenden Befehl umgesetzt werden:

```
# Redis systemd-Dienstdatei aktivieren
systemctl enable redis-server.service
```

A80 Die Systemzeit muss über das Network Time Protocol synchronisiert werden.

Für Neo4j und Redis ist kein integrierter NTP-Dienst/Mechanismus vorhanden. Stattdessen kann die Synchronisation der Systemzeit mittels `systemd-timesyncd` erfolgen und ist in der Datei /etc/systemd/timesyncd.conf wie folgt zu konfigurieren:

```
# systemd-timesyncd konfigurieren
[Time]
NTP=0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org 3.pool.ntp.org
FallbackNTP=ntp.ubuntu.com 0.ubuntu.pool.ntp.org
↪ 1.ubuntu.pool.ntp.org 2.ubuntu.pool.ntp.org 3.ubuntu.pool.ntp.org
RootDistanceMaxSec=1
PollIntervalMinSec=32
PollIntervalMaxSec=2048
ConnectionRetrySec=30
SaveIntervalSec=60
```

Anschließend muss die zugehörige Systemd-Dienstdatei mit dem folgenden Befehl aktiviert werden:

```
# systemd-timesyncd-Dienstdatei aktivieren
systemctl enable systemd-timesyncd.service
```

4.1.12 Backup und Replikation

A81 Es müssen regelmäßige Systemsicherungen des Datenbanksystems durchgeführt werden.

Neo4j Neo4j verfügt in der Community-Edition nicht über eine integrierte Funktion zur automatischen Erstellung von Datenbanksicherungen. Die export-APOC-Prozeduren können jedoch manuell verwendet werden, um eine Sicherung der Datenbanken zu erstellen.

Redis Redis verfügt in der Community-Edition nicht über eine integrierte Funktion zur automatischen Erstellung von Datenbanksicherungen.

A82 Für die Durchführung von Systemsicherungen muss ein eigenständiger Benutzer verwendet werden.

Für Neo4j und Redis kann dies nicht umgesetzt werden, da es für beide Datenbanksysteme in der Community-Edition keine integrierte Funktion zur automatischen Erstellung von Datenbanksicherungen gibt. Dies ist jedoch die Grundvoraussetzung für die Umsetzung dieser Anforderung.

4.2 Implementierung des InSpec Compliance-Profils für Redis

Im Zuge der Anwendung des Vorgehensmodells im vorherigen Abschnitt 4.1 wurden die konkreten technischen Maßnahmen zur Realisierung einer gehärteten Konfiguration für beide Datenbanksysteme bewusst sehr ausführlich dargestellt. Dadurch wird die Überführung dieser Maßnahmen in ein InSpec-Compliance-Profil (siehe auch Abschnitt 2.3) wesentlich erleichtert, da die dafür notwendigen Vorbereitungsarbeiten, unter anderem in Form der Identifikation aller relevanten Konfigurationsoptionen und deren sichere Sollwerte, bereits umfassend vorliegen. Die Implementierung wurde für Redis 7.x durchgeführt und gewährleistet eine Kompatibilität mit Debian-basierten Derivaten. Diese wurde auf GitHub unter `https://github.com/lonkey/redis-baseline` mit zwei Verzweigungen (englisch „Branch“) veröffentlicht:

main Die Verzweigung wird nach Abgabe der Master-Thesis genutzt, um im Rahmen einer Weiterentwicklung die Kompatibilität sowohl zu weiteren Redis-Versionen als auch zu weiteren Linux-Distributionen zu verwirklichen.

masterthesis Die Verzweigung wird nach Abgabe der Master-Thesis eingefroren, um den Entwicklungsstand auf der Basis der hier gesetzten Rahmenbedingungen zu erhalten.

Im Rahmen der Implementierung des Compliance-Profils wurde auf mehrere Standard-InSpec-Ressourcen zurückgegriffen, um beispielsweise Dateiberechtigungen zu prüfen, die Linux-Audit-Deamon-Konfiguration zu validieren oder sicherzustellen, dass die zugehörigen systemd-Dienstdateien ordnungsgemäß konfiguriert und aktiv sind. Für die Auswertung der Redis-Konfiguration war es hingegen erforderlich, eine entsprechende Ressourcenerweiterung selbst zu entwickeln, da in InSpec keine integrierte Ressource zur Interpretation der Redis-Konfigurationssyntax zur Verfügung steht. Diese befindet sich in der Datei /libraries/redis_conf.rb und ermöglicht es InSpec in erster Linie, die Struktur der Redis-Konfigurations- und ACL-Benutzerdatei mithilfe des regulären Ausdrucks aus Zeile 66 des Listings 5 zu verstehen:

```
61 def read_params
62   return @params if defined?(@params)
63
64   conf = SimpleConfig.new(
65     @content,
66     assignment_regex: /^\s*(\S+)\s+(.*)\s*$/,
67     multiple_values: true
68   )
```

```
69   @params = conf.params
70 end
```

Listing 5: Funktion „read_params“ aus der selbstentwickelten Ressourcenerweiterung für Redis

Die Syntax der Bezeichnungen der Konfigurationsoptionen wird in der ersten Erfassungsgruppe (`(\S+)`) und die Syntax der Konfigurationswerte in der zweiten Erfassungsgruppe (`(.*)`) definiert. In der Mitte werden die beiden Erfassungsgruppen durch ein Trennzeichen (`\s+`), in diesem Fall in Form eines Leerzeichens, voneinander separiert. Schließlich wird durch die Option `multiple_values: true` in Zeile 67 bestimmt, dass eine Konfigurationsoption grundsätzlich mehrere Werte enthalten kann.

Das Verhalten eines InSpec-Profils kann grundsätzlich durch Variablen an abweichende Ausführungsumgebungen angepasst werden. Die hier vorgestellte Implementierung der Redis 7.x Community-Edition Baseline erlaubt die Anpassung von Daten- und Konfigurationspfaden, Bezeichnungen von Konfigurationsdateien sowie der Namen der Benutzernamen und -gruppen zur Laufzeit über Kommandozeilenargumente. Diese werden in der Datei /controls/redis_conf.rb in Form von InSpec-Eingaben (englisch „Inputs“) definiert, wie nachfolgend in Listing 6 dargestellt:

```
22 [...]
23 redis_custom_conf_dir = input('redis_custom_conf_dir', value: '/etc/redis',
   ↪ description: 'The Redis configuration files may be located in a
   ↪ different directory')
24 redis_custom_data_dir = input('redis_custom_data_dir', value:
   ↪ '/var/lib/redis', description: 'The Redis database files may be located
   ↪ in a different directory')
25 redis_custom_acl_file = input('redis_custom_acl_file', value: 'users.acl',
   ↪ description: 'The Redis ACL file may have a different name')
26 redis_custom_conf_file = input('redis_custom_conf_file', value:
   ↪ 'redis.conf', description: 'The Redis configuration file may have a
   ↪ different name')
```

Listing 6: Definierung der InSpec-Eingaben

Anschließend wird die eigenentwickelte Redis-Ressourcenerweiterung innerhalb derselben Datei verwendet, um mit den universellen Fitern (englisch „Matchers“) von

InSpec direkt auf einzelne Konfigurationsoptionen und deren -werte zuzugreifen. Ein gutes Beispiel hierfür ist die in Listing 7 zu sehende Implementierung der Anforderung A11:

```
131 control 'redis-a11' do
132   impact 1.0
133   title 'Sind mehrere Authentifizierungsmechanismen verfügbar, ist das
      ↪  sicherste Verfahren zu verwenden'
134   desc 'ACL-Datei verwenden'
135   describe redis_conf("#{redis_conf_file}") do
136     its('aclfile') { should eq "#{redis_acl_file}" }
137   end
138   describe redis_conf("#{redis_acl_file}") do
139     its('content') { should_not match(/^(?:(?!user\s).)+$/) }
140   end
141 end
```

Listing 7: Implementierung der InSpec-Kontrolle von Anforderung A11

Zunächst wird in Zeile 136 die Ressourcenerweiterung eingesetzt, um zu prüfen, ob der Wert der Konfigurationsoption `aclfile`, die zur Festlegung des Dateipfads der ACL-Benutzerdatei genutzt wird, mit der Variable `redis_acl_file` übereinstimmt. In Zeile 139 wird dann mit Hilfe eines regulären Ausdrucks sichergestellt, dass jede Zeile der Benutzerdatei nur mit dem Präfix `user` beginnt, um auszuschließen, dass fälschlicherweise andere Konfigurationsoptionen in der Datei enthalten sind.

Ein weiteres gutes Beispiel zur Darstellung eines anderen Anwendungszwecks ist die Implementierung der Anforderung A35 in Listing 8:

```
226 control 'redis-a35' do
227   impact 1.0
228   title 'Passwörter und kryptografische Schlüssel dürfen nur einen einzigen
      ↪  Einsatzzweck aufweisen und nicht mehrfach verwendet werden'
229   desc 'Suche nach Passwort-Hash-Duplikaten'
230   describe command("grep -oe \"#[0-9a-f]\\{64\\}\" #{redis_acl_file} | uniq
      ↪  -c") do
231     its('stdout') { should match(/^\s*1\s.*$/) }
232     its('stderr') { should eq '' }
233     its('exit_status') { should eq 0 }
234   end
235 end
```

Listing 8: Implementierung der InSpec-Kontrolle von Anforderung A35

In Zeile 220 wird die ACL-Benutzerdatei auf doppelte SHA-256 Hash-Werte geprüft, indem ein lokaler Befehl systemseitig ausgeführt wird, um Passwort-Duplikate zu identifizieren. In Zeile 221 wird die Ausgabe dann mit einem regulären Ausdruck evaluiert. Schließlich wird in den Zeilen 212 und 213 die Standardfehlerausgabe untersucht, um sicherzustellen, dass die Ausführung des Befehls ordnungsgemäß erfolgt ist.

Dieser Vorgehensweise wurde für alle in Abschnitt 4.1 genannten Anforderungen verfolgt, sodass bis auf wenige Ausnahmen entweder ein einfacher Soll-Ist-Vergleich oder eine komplexere Analyse des Konfigurationswerts mithilfe eines regulären Ausdrucks durchgeführt wird. Das Ergebnis der Ausführung des InSpec Compliance-Profils für Redis kann dem folgenden Bild 1 entnommen werden:

```
root@redis: ~
  ✔  redis-a76: Die Datenbankanwendung darf nicht an 0.0.0.0 bzw. [::] gebunden werden
     ✔  redis.conf bind is expected not to match /0\.0\.0\.0(\s|$)/
     ✔  redis.conf bind is expected not to match /::(\s|$)/
  ✔  redis-a77: Die Datenbankanwendung darf nicht an den Standard-Port gebunden werden
     ✔  redis.conf port is expected not to eq "6379"
     ✔  redis.conf tls-port is expected not to eq "6379"
  ×  redis-a79: Die Systemd-Dienstdateien müssen aktiviert werden (2 failed)
     ✔  Service redis-server is expected to be installed
     ×  Service redis-server is expected to be enabled
     expected that `Service redis-server` is enabled
     ×  Service redis-server is expected to be running
     expected that `Service redis-server` is running
  ✔  redis-a80: Die Systemzeit muss über das Network Time Protocol synchronisiert werden
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^NTP=0\.pool\.ntp\.org\s1\.p
ool\.ntp\.org\s2\.pool\.ntp\.org\s3\.pool\.ntp\.org$/
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^FallbackNTP=ntp\.ubuntu\.co
m\s0\.ubuntu\.pool\.ntp\.org\s1\.ubuntu\.pool\.ntp\.org\s2\.ubuntu\.pool\.ntp\.org\s3\.ubuntu\.pool\.ntp\.org$/
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^RootDistanceMaxSec=1$/
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^PollIntervalMinSec=32$/
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^PollIntervalMaxSec=2048$/
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^ConnectionRetrySec=30$/
     ✔  Parse Config File /etc/systemd/timesyncd.conf content is expected to match /^SaveIntervalSec=60$/
     ✔  Service systemd-timesyncd is expected to be installed
     ✔  Service systemd-timesyncd is expected to be enabled
     ✔  Service systemd-timesyncd is expected to be running


Profile Summary: 32 successful controls, 5 control failures, 0 controls skipped
Test Summary: 350 successful, 8 failures, 0 skipped
root@redis:~#
```

Bild 1: Ausgabe des InSpec Compliance-Profils für Redis

Die Ergebnisse werden beginnend mit der Überprüfung der Implementierung der Anforderung A76 angezeigt und entsprechend ihrem Konformitätsstatus farblich grün oder rot hervorgehoben. Für jede Anforderung wird die korrekte Umsetzung der gesamten Kontrolle angezeigt, die aus mehreren einzelnen Unterkontrollen bestehen kann, erkennbar an der eingerückten Darstellung. Abschließend folgt die Ausgabe der Statistik zur Konformitätsrate aller Kontrollen als Ganzes sowie aller insgesamt vorhandenen Unterkontrollen. Im vorliegenden Beispiel sind 32 von 37 Kontrollen vollständig erfüllt und von den insgesamt 358 Unterkontrollen, aus denen sich die 37 Kontrollen zusammensetzen, werden 350 als erfüllt eingestuft.

5 Auswertung der Anwendung des Vorgehensmodells

In Abschnitt 4.1 wurden anhand des erstellten Vorgehensmodells die Datenbanksysteme Neo4j und Redis auf die Ableitung von möglichen technischen Härtungsmaßnahmen untersucht. Zu diesem Zweck sind für jede Anforderung konkrete Maßnahmen festgehalten oder, falls diese unter Berücksichtigung des durch das Vorgehensmodell gesetzten Rahmens nicht oder nur teilweise umsetzbar waren, ein entsprechender Hinweis auf Nichterfüllung formuliert worden. In diesem Kapitel wird der Erfüllungsgrad für beide Datenbanksysteme ermittelt. Zur Ermöglichung einer feingranularen Auswertung wird der Erfüllungsgrad in Abschnitt 5.1 zunächst für jede der 12 Anforderungskategorien einzeln und in Abschnitt 5.2 schließlich für alle 82 Anforderungen des Vorgehensmodells insgesamt berechnet. Abschließend werden in Abschnitt 5.3 verschiedene offene Forschungsfragen, die sich bei der Auseinandersetzung mit dem Vorgehensmodell ergeben können, erläutert und eingeordnet.

5.1 Ermittlung des Erfüllungsgrades nach Anforderungskategorien

Zur Ermittlung des Erfüllungsgrades werden die Checklisten des IT-Grundschutz-Kompendiums als Orientierung herangezogen [94]. Der in den Checklisten enthaltene Erfüllungsgrad „entbehrlich“ wurde in die Auswertung nicht einbezogen, da das Vorgehensmodell speziell für den Einsatz bei nicht-relationalen Datenbanksystemen entwickelt wurde, sodass grundsätzlich alle Anforderungen zu berücksichtigen sind. Die einzige Ausnahme bildet die Anforderung A1 aufgrund ihrer besonderen Stellung im Demingkreis für die Weiterentwicklung des Vorgehensmodells, die bereits im Vorwort zu Abschnitt 3.2 erläutert wurde. Den verbleibenden drei Erfüllungskategorien wurde anschließend ein Punktwert zugewiesen, um den Erfüllungsgrad prozentual darstellen zu können und somit auch eine bessere Vergleichbarkeit zu schaffen. Die Erfüllungskategorien sind in der folgenden Tabelle 4 entsprechend ihrer anschließenden Verwendung dargestellt:

Erfüllungsgrad	Punktwert	Beschreibung
Ja	2	Die Anforderung kann durch das DBMS mittels geeigneter Maßnahmen vollständig und effektiv umgesetzt werden.
Teilweise	1	Die Anforderung kann durch das DBMS nicht vollständig oder nicht mit der erforderlichen Wirksamkeit umgesetzt werden.
Nein	0	Die Anforderung kann durch das DBMS nicht umgesetzt werden.

Tabelle 4: Für die Auswertung herangezogene Erfüllungskategorien

In den nachfolgenden Tabellen wird nun allen Anforderungen der jeweilige Punktwert zur Ermittlung des Erfüllungsgrades zugeordnet. Anschließend folgt für jede Tabelle eine Zwischenauswertung für beide Datenbanksysteme in Form der Angabe des erreichten Punktwertes und des daraus errechneten prozentualen Erfüllungsgrades innerhalb der Anforderungskategorie.

Grundprinzipien des Vorgehensmodells

Anforderung	Bezeichnung	Neo4j	Redis
A1	Die Standardwerte aller sicherheitsrelevanten Konfigurationsparameter müssen explizit festgelegt werden.	-	-
A2	Nicht benötigte Plug-ins/Software-Erweiterungen und Funktionen müssen deinstalliert oder deaktiviert werden.	2	2

Tabelle 5: Auswertung der Anforderungskategorie „Grundprinzipien des Vorgehensmodells“

Für die in Tabelle 5 dargestellte Anforderungskategorie „Grundprinzipien des Vorgehensmodells“ können unter Berücksichtigung der Ausklammerung der Anforderung A1 maximal 2 Punktwerte erreicht werden. Für Neo4j und Redis wurden insgesamt jeweils 2 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von jeweils $100,00\ \%$ entspricht.

Installation und Updates

Anforderung	Bezeichnung	Neo4j	Redis
A3	Der Aktualisierungsmechanismus muss sicher konfiguriert werden.	2	2
A4	Die Herkunft der Software-Installations- und -Aktualisierungspakete aus vertrauenswürdigen Quellen muss gewährleistet werden.	2	2
A5	Die Herkunft von Plug-ins/Software-Erweiterungen aus vertrauenswürdigen Quellen muss gewährleistet werden.	1	1
A6	Die Integrität der Software-Installations- und -Aktualisierungspakete muss verifiziert werden.	2	2
A7	Die Integrität der Plug-ins/Software-Erweiterungen muss verifiziert werden.	1	0
A8	Die Version der Software und der Plug-ins/Software-Erweiterungen müssen vom Hersteller unterstützt werden.	2	2
A9	Die Installation von Aktualisierungspaketen muss zeitnah erfolgen.	1	1

Tabelle 6: Auswertung der Anforderungskategorie „Installation und Updates“

Für die in Tabelle 6 dargestellte Anforderungskategorie „Installation und Updates“ können maximal 14 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 11 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 78,57$ % entspricht. Für Redis wurden insgesamt 10 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 71,43$ % entspricht.

Authentifizierung

Anforderung	Bezeichnung	Neo4j	Redis
A10	Die Authentifizierung muss konfiguriert und aktiv sein.	2	2
A11	Sind mehrere Authentifizierungsmechanismen verfügbar, ist das sicherste Verfahren zu verwenden.	0	1
A12	Die Authentifizierung aller Teilnehmer in einem Cluster muss konfiguriert und aktiv sein.	0	0
A13	Die Authentifizierung muss an sämtlichen Schnittstellen/Interfaces erfolgen.	2	2
A14	Die Authentifizierung muss, wenn möglich, mehrere Authentifizierungsmerkmale umfassen (Multi-Faktor-Authentifizierung).	2	2
A15	Sitzungskennungen müssen zufällig erzeugt werden und dürfen kein vorhersehbares Schema aufweisen.	2	2
A16	Fehlgeschlagene Authentifizierungen dürfen nicht zur Durchführung von Angriffen interpretiert werden können.	2	2
A17	Wenn mehrere Authentifizierungsversuche fehlschlagen, müssen Trigger definiert werden, um weitere Versuche zu verzögern.	2	0
A18	Wenn mehrere Authentifizierungsversuche fehlschlagen, müssen Trigger definiert werden, um aktive Sitzungen zu beenden oder Benutzer zu sperren.	0	0
A19	Die Gesamtdauer eines Anmeldeversuchs muss begrenzt werden.	0	0
A20	Die Anzahl der gleichzeitigen Verbindungen zur Datenbank muss begrenzt werden.	0	2
A21	Die Anzahl der parallel aktiven Sitzungen pro Benutzer muss begrenzt werden.	0	2
A22	Die Zwischenspeicherung von Authentifizierungsdaten muss deaktiviert werden.	2	2

Tabelle 7: Auswertung der Anforderungskategorie „Authentifizierung“

Für die in Tabelle 7 dargestellte Anforderungskategorie „Authentifizierung“ können maximal 26 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 14 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 53,85$ % entspricht. Für Redis wurden insgesamt 17 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 65,38$ % entspricht.

Autorisierung

Anforderung	Bezeichnung	Neo4j	Redis
A23	Die Autorisierung muss konfiguriert und aktiv sein.	2	2
A24	Jeder Benutzer muss einer Berechtigungsgruppe/-Access Control List zugewiesen sein.	1	1
A25	Benutzerkonten, die über einen längeren Zeitraum inaktiv sind, müssen deaktiviert werden.	0	0
A26	Benutzerkonten und -gruppen, die deaktiviert sind/nicht verwendet werden, müssen gelöscht werden.	0	0
A27	Die vordefinierten Benutzerrollen sind so weit wie möglich zu verwenden.	2	0
A28	Die vordefinierten Benutzerrollen sind auf ihre Vereinbarkeit im Hinblick auf alle Anforderungen zu prüfen.	0	0
A29	Die gleiche Benutzerkennung darf nicht von mehreren Personen oder Diensten verwendet werden.	0	0
A30	Die gleiche Benutzerkennung darf nicht für den Zugriff auf mehrere Datenbanken verwendet werden.	0	2
A31	Eine rollenbasierte Zugriffskontrolle zur Trennung von Benutzer- und Datenbankverwaltungsfunktionen muss umgesetzt werden.	0	2
A32	Die Vergabe von Zugriffsrechten muss nach dem Least-Privilege- und Erforderlichkeitsprinzip erfolgen.	0	2

Tabelle 8: Auswertung der Anforderungskategorie „Autorisierung“

Für die in Tabelle 8 dargestellte Anforderungskategorie „Authentifizierung“ können maximal 20 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 5 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $25,00$ % entspricht. Für Redis wurden insgesamt 9 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $45,00$ % entspricht.

Passwortrichtlinien

Anforderung	Bezeichnung	Neo4j	Redis
A33	Verwendete Passwörter müssen hohen Sicherheitsanforderungen standhalten.	0	0
A34	Vordefinierte Standard-Passwörter und -Benutzerkennungen dürfen nicht verwendet werden.	2	2
A35	Passwörter und kryptografische Schlüssel dürfen nur einen einzigen Einsatzzweck aufweisen und nicht mehrfach verwendet werden.	0	2
A36	Frühere Passwörter dürfen nicht wiederverwendet werden.	0	0
A37	Passwörter dürfen nur mit einer sicheren Methode als Hash unter Verwendung eines Salts sowie, falls möglich, mit Peppering gespeichert werden.	2	1
A38	Mechanismen zum Zurücksetzen von Passwörtern dürfen keine Angriffsfläche für Angreifer bieten.	2	2
A39	Passwörter dürfen nicht aufgrund von zeitlichen Nutzungsbegrenzungen geändert werden.	2	2
A40	Zur Erkennung von Passwortkompromittierungen müssen geeignete Schutzmaßnahmen ergriffen werden.	1	1

Tabelle 9: Auswertung der Anforderungskategorie „Passwortrichtlinien“

Für die in Tabelle 9 dargestellte Anforderungskategorie „Passwortrichtlinien“ können maximal 16 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 9 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $56,25$ % entspricht. Für Redis wurden insgesamt 10 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $62,50$ % entspricht.

Auditierung und Protokollierung

Anforderung	Bezeichnung	Neo4j	Redis
A41	Die Auditierung sowie Protokollierung muss konfiguriert und aktiv sein.	2	2
A42	Sind mehrere Audit- und Protokollierungsfunktionen verfügbar, ist das sicherste Verfahren zu verwenden.	1	2
A43	Für Audit-Protokolle muss ausreichend Speicherplatz bereitgestellt werden.	2	2
A44	Audit-Protokolle müssen in ein separates Log-Management-System ausgelagert werden.	1	1
A45	Um den Verlust von Audit-Protokollen zu verhindern, müssen Warnungen gesendet werden, wenn der Speicherplatz knapp wird oder die Protokollierung fehlschlägt.	1	1
A46	Audit-Protokolle sind geordnet nach ihrem Alter zu überschreiben, wenn der Speicherplatz für neue Einträge erschöpft ist.	0	0
A47	Audit-Protokolle müssen alle Ereignisse und Aktivitäten erfassen (maximale Verbosität).	2	2
A48	Audit-Protokolle müssen einem vordefinierten Format entsprechen, das ihre Analyse erleichtert.	2	2
A49	Audit-Protokolle müssen in einem Verzeichnis mit leicht zuzuordnenden Dateinamen gespeichert werden.	2	2
A50	Der Zugriff auf die Konfiguration der Auditierung und Protokollierung muss begrenzt werden.	2	2
A51	Der Zugriff auf die Inhalte der Auditierung und Protokollierung muss begrenzt werden.	2	2

Tabelle 10: Auswertung der Anforderungskategorie „Auditierung und Protokollierung“

Für die in Tabelle 10 dargestellte Anforderungskategorie „Auditierung und Protokollierung“ können maximal 22 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 17 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von

$\approx 77,27$ % entspricht. Für Redis wurden insgesamt 18 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 81,82$ % entspricht.

Monitoring

Anforderung	Bezeichnung	Neo4j	Redis
A52	Alle kritischen Parameter, Ereignisse und Betriebszustände müssen überwacht werden.	1	1

Tabelle 11: Auswertung der Anforderungskategorie „Monitoring“

Für die in Tabelle 11 dargestellte Anforderungskategorie „Monitoring“ können maximal 2 Punktwerte erreicht werden. Für Neo4j und Redis wurde insgesamt jeweils 1 Punktwert vergeben, was einem prozentualen Erfüllungsgrad von jeweils $50,00$ % entspricht.

Fingerprinting

Anforderung	Bezeichnung	Neo4j	Redis
A53	Der Debug-Modus muss deaktiviert werden.	2	2
A54	Die Ausgabe von Fehlermeldungen darf nicht zur Durchführung von Angriffen interpretiert werden können.	2	2
A55	Die Verbindung zum Datenbanksystem darf keine Rückschlüsse auf die Version zulassen.	2	2

Tabelle 12: Auswertung der Anforderungskategorie „Fingerprinting“

Für die in Tabelle 12 dargestellte Anforderungskategorie „Fingerprinting“ können maximal 6 Punktwerte erreicht werden. Für Neo4j und Redis wurden insgesamt jeweils 6 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von jeweils $100,00$ % entspricht.

Verschlüsselung

Anforderung	Bezeichnung	Neo4j	Redis
A56	Die Kommunikation über Schnittstellen muss verschlüsselt erfolgen.	2	2
A57	Die Kommunikation aller Teilnehmer in einem Cluster muss verschlüsselt erfolgen.	0	0
A58	Die Verschlüsselung muss mit sicheren kryptografischen Protokollen betrieben werden.	2	2
A59	Das für den Schlüsselaustausch verwendete Verfahren muss sicher sein.	2	2
A60	Die Verschlüsselung muss mit sicheren kryptografischen Algorithmen betrieben werden.	2	2
A61	Die kryptografischen Algorithmen müssen eine hohe Schlüssellänge aufweisen.	2	2
A62	Selbstsignierte Zertifikate dürfen für eine verschlüsselte Kommunikation nicht verwendet und akzeptiert werden.	2	2
A63	Für die Erstellung von Anmeldeinformationen und Zertifikaten müssen sichere Schlüsselgeneratoren verwendet werden.	2	1
A64	Die Anwendungsdaten müssen verschlüsselt werden.	0	0

Tabelle 13: Auswertung der Anforderungskategorie „Verschlüsselung“

Für die in Tabelle 13 dargestellte Anforderungskategorie „Verschlüsselung“ können maximal 18 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 14 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 77,78$ % entspricht. Für Redis wurden insgesamt 13 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 72,22$ % entspricht.

Verzeichnis- und Dateiberechtigungen

Anforderung	Bezeichnung	Neo4j	Redis
A65	Die Zugriffsrechte auf zur Datenbankanwendung gehörende Verzeichnisse, Dateien und Anwendungen müssen restriktiv vergeben werden.	2	2
A66	Die Zugriffsrechte auf Protokollierungsdaten müssen restriktiv vergeben werden.	2	2
A67	Die Zugriffsrechte auf kryptografische Schlüssel müssen restriktiv vergeben werden.	2	2

Tabelle 14: Auswertung der Anforderungskategorie „Verzeichnis- und Dateiberechtigungen“

Für die in Tabelle 14 dargestellte Anforderungskategorie „Verzeichnis- und Dateiberechtigungen“ können maximal 6 Punktwerte erreicht werden. Für Neo4j und Redis wurden insgesamt jeweils 6 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von jeweils $100,00$ % entspricht.

Sicherer Betrieb der Datenbankanwendung

Anforderung	Bezeichnung	Neo4j	Redis
A68	Datenbankspezifische Schutzmechanismen müssen konfiguriert und aktiviert werden.	0	0
A69	Funktionen, die die Ausführung von dynamischem Code verhindern, müssen aktiviert werden.	2	0
A70	Die Ausführung von Datenbank-Skripten muss deaktiviert werden, oder die Skripte müssen umfassend auf Schwachstellen geprüft werden.	2	2
A71	Die verfügbaren Systemressourcen müssen für den Datenbankbetrieb optimiert werden.	0	0
A72	Die Datenbank muss erfolgreich initialisiert werden.	0	0
A73	Die Datenbank muss in einen stabilen Zustand übergehen, sollte die Initialisierung fehlschlagen.	0	2

A74	Die Datenbankanwendung muss unter eigenem Benutzer und eigener Gruppe ausgeführt werden.	2	2
A75	Die Datenbankanwendung muss mit möglichst geringen Berechtigungen ausgeführt werden.	2	2
A76	Die Datenbankanwendung darf nicht an 0.0.0.0 bzw. [::] gebunden werden.	2	2
A77	Die Datenbankanwendung darf nicht an den Standard-Port gebunden werden.	2	2
A78	Die Management-Schnittstelle muss sich in einem dedizierten Netzwerksegment befinden und der Zugriff begrenzt werden.	2	0
A79	Die Systemd-Dienstdateien müssen aktiviert werden.	2	2
A80	Die Systemzeit muss über das Network Time Protocol synchronisiert werden.	2	2

Tabelle 15: Auswertung der Anforderungskategorie „Sicherer Betrieb der Datenbankanwendung“

Für die in Tabelle 15 dargestellte Anforderungskategorie „Sicherer Betrieb der Datenbankanwendung“ können maximal 26 Punktwerte erreicht werden. Für Neo4j wurden insgesamt 18 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 69,23$ % entspricht. Für Redis wurden insgesamt 16 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 61,54$ % entspricht.

Backup und Replikation

Anforderung	Bezeichnung	Neo4j	Redis
A81	Es müssen regelmäßige Systemsicherungen des Datenbanksystems durchgeführt werden.	1	0
A82	Für die Durchführung von Systemsicherungen muss ein eigenständiger Benutzer verwendet werden.	0	0

Tabelle 16: Auswertung der Anforderungskategorie „Backup und Replikation“

Für die in Tabelle 16 dargestellte Anforderungskategorie „Backup und Replikation“ können maximal 4 Punktwerte erreicht werden. Für Neo4j wurde insgesamt 1 Punktwert vergeben, was einem prozentualen Erfüllungsgrad von $25,00$ % entspricht. Für Redis wurden keine Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $0,00$ % entspricht.

5.2 Ermittlung des ganzheitlichen Erfüllungsgrades

Zur Ermittlung des ganzheitlichen Erfüllungsgrades werden die für alle 82 Anforderungen vergebenen Punktwerte summiert und der absolute prozentuale Erfüllungsgrad berechnet. Für den bestmöglichen Überblick über alle Ergebnisse werden außerdem zunächst die vorherigen Einzelergebnisse aller 12 Anforderungskategorien noch einmal vorangestellt.

Anforderungskategorie	**Punktwerte**		**Prozentwerte**	
	Neo4j	**Redis**	**Neo4j**	**Redis**
Grundprinzipien des Vorgehensmodells	2 / 2	2 / 2	$100,00$ %	$100,00$ %
Installation und Updates	11 / 14	10 / 14	$\approx 78,57$ %	$\approx 71,43$ %
Authentifizierung	14 / 26	17 / 26	$\approx 53,85$ %	$\approx 65,38$ %
Autorisierung	5 / 20	9 / 20	$25,00$ %	$45,00$ %
Passwortrichtlinien	9 / 16	10 / 16	$56,25$ %	$62,50$ %
Auditierung und Protokollierung	17 / 22	18 / 22	$\approx 77,27$ %	$\approx 81,82$ %
Monitoring	1 / 2	1 / 2	$50,00$ %	$50,00$ %
Fingerprinting	6 / 6	6 / 6	$100,00$ %	$100,00$ %
Verschlüsselung	14 / 18	13 / 18	$\approx 77,78$ %	$\approx 72,22$ %
Verzeichnis- und Dateiberechtigungen	6 / 6	6 / 6	$100,00$ %	$100,00$ %
Sicherer Betrieb der Datenbankanwendung	18 / 26	16 / 26	$\approx 69,23$ %	$\approx 61,54$ %

Backup und Replikation	1 / 4	0 / 4	$25,00$ %	$0,00$ %
Zusammenfassung	**104 / 162**	**108 / 162**	$\approx$ **64,20 %**	$\approx$ **66,67 %**

Tabelle 17: Zusammenfassung der Auswertung aller Anforderungskategorien und Ermittlung des ganzheitlichen Erfüllungsgrades

In Tabelle 17 werden die Erfüllungsgrade für jede der 12 Anforderungskategorien und abschließend, in der letzten fett hervorgehobenen Zeile, für alle 82 Anforderungen insgesamt zusammenfassend dargestellt. Für Neo4j wurden insgesamt 104 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 64,20$ % entspricht. Für Redis wurden insgesamt 108 Punktwerte vergeben, was einem prozentualen Erfüllungsgrad von $\approx 66,67$ % entspricht.

5.3 Weiterführende Evaluierungsfragen und Einordnungen

Im Laufe der Ausarbeitung dieser Master-Thesis ergaben sich eine Reihe weiterer interessanter Fragestellungen, die an dieser Stelle kurz angesprochen und eingeordnet werden.

1. **Hat die Anwendung des Vorgehensmodells im Vergleich zum bestehenden STIGs-Leitfaden zusätzliche technische Maßnahmen für Redis ergeben?**

   Diese Fragestellung ist insbesondere deshalb besonders interessant, da so die Effektivität und somit die Ganzheitlichkeit des Vorgehensmodells auf sehr einfache und effiziente Art und Weise hätte evaluiert werden können. Schlussendlich war es jedoch nicht möglich, eine entsprechende Auswertung in Form einer Gegenüberstellung der Anforderungen/Maßnahmen beider Modelle durchzuführen. Dies ist auf folgende Faktoren zurückzuführen:

   Unzureichende Vergleichbarkeit Der vorhandene „Redis Enterprise 6.x Security Technical Implementation Guide“ beleuchtet die Absicherung der Enterprise-Edition von Redis, während im Rahmen dieser Master-Thesis die frei verfügbare Community-Edition zum Einsatz kam. Zwischen den beiden Editionen besteht ein sehr großer Unterschied in den verfügbaren Funktionen, insbesondere im Hinblick auf Sicherheitsfunktionalitäten im Zusammenhang mit ACLs, Monitoring sowie dem Betrieb von Clustern. Gerade die ersten beiden Funktionalitäten sind für eine Auswertung des Vorgehensmodells besonders interessant. Sie eignen sich jedoch nicht für diesen Zweck, da zum einen die vorhandenen Funktionen nicht nur nicht identisch sind, sondern sich auch in der Art und Weise ihrer Implementierung stark unterscheiden. Weiterhin wird in der STIGs die Version 6.x von Redis betrachtet, während in der vorliegenden Arbeit die neuere Version 7.x analysiert wurde.

   Unzureichende Kompatibilität Nicht nur die verfügbaren Sicherheitsfunktionalitäten unterscheiden sich stark, sondern auch die verfügbaren Schnittstellen. Im Gegensatz zur Redis Community-Edition, die nur eine zentrale Schnittstelle sowohl für die Administration als auch für den Datenzugriff aufweist, verfügt die Enterprise-Edition über eine Vielzahl von Schnittstellen, unter anderem für den Discovery-Service, für die REST-API und den Web-Proxy, für die Internode-Kommunikation oder auch zur Bereitstellung der webbasierten Management-Schnittstelle. Aus diesem Grund können die umzusetzenden Sicherheitsmaßnah-

men zwischen den beiden Editionen nicht zueinander in Beziehung gesetzt werden, da sowohl unterschiedliche Sicherheitsanforderungen und Konfigurationsoptionen erforderlich sind als auch unterschiedliche Werkzeuge bzw. Schnittstellen zu deren Realisierung verwendet werden.

Abweichender Geltungsbereich Die STIGs bewerten das Produkt anhand der Cybersicherheitsanforderungen des DoDs [95]. Dies beinhaltet auch eine Reihe von Sicherheitsmaßnahmen, die nur durch organisatorische, prozedurale Maßnahmen realisiert werden können, während der Fokus des vorliegenden Vorgehensmodells allein auf der Ergreifung von technischen Maßnahmen liegt. Dieser abweichende Geltungsbereich reduziert die Vergleichsmöglichkeiten zusätzlich.

Zusammenfassend war es daher nicht möglich, eine Auswertung durchzuführen, die einen Informationsgewinn dargestellt hätte. Alle Maßnahmen, die für beide Editionen in vergleichbarer Weise ergriffen werden können, wurden im Zuge der Anwendung des Vorgehensmodells in Abschnitt 4.1 vollumfänglich umgesetzt. Eine alternative Betrachtung für Neo4j war ebenfalls nicht möglich, da weder ein CIS-Benchmark noch ein STIGs-Leitfaden für das Datenbanksystem zur Verfügung steht.

2. **Aus welchem Grund lassen sich alle bisher identifizierten Anforderungen auch auf relationale Datenbanksysteme übertragen/anwenden?**

   Mit Ausnahme des systemorientierten Bausteins „APP.4.3: Relationale Datenbanksysteme“ aus dem BSI IT-Grundschutz-Kompendium basiert das in Kapitel 3 entwickelte Vorgehensmodell ausschließlich auf jenen CIS-Benchmarks und STIGs-Leitfäden, die sich mit der Härtung nicht-relationaler Datenbanksysteme befassen. Eine nicht ausreichend zielgerichtete Entwicklung bzw. Arbeitsgrundlage für die Schaffung des Vorgehensmodells ist daher als Ursache eher unwahrscheinlich. Vielmehr liegt der Ursprung dieses Effekts in dem Umstand, dass sich relationale und nicht-relationale Datenbanksysteme aus einer Sicherheits- und somit auch aus einer Härtungsperspektive kaum unterscheiden. Letztlich handelt es sich bei beiden Typen von Datenbanksystemen um Systeme, die der Speicherung, Verarbeitung und Bereitstellung von Informationen dienen und zu diesem Zweck eine klar definierte Zusammenstellung bzw. Verknüpfung von Datensätzen oder Dateien beinhalten. Der größte Unterschied liegt also in der Art und Weise, wie die Daten abgespeichert werden, und in der Technologie, die zur Interaktion mit diesen zur Anwendung kommt. An diesem Punkt hätten sich dementsprechend durchaus Unterschiede ergeben können, z. B. im Rahmen der Umsetzung

der Anforderungen A69 und A70, die sich mit der Absicherung der jeweiligen Abfragesprache und der Ausführung von Datenbank-Skripten beschäftigen. Dies ist jedoch nicht der Fall, da bei einer übergeordneten/gesamtheitlichen Betrachtung das Risikopotenzial in Form von Code-Injektionen oder der Ausführung von bösartigen Skripten unverändert bleibt. Daher müssen die entsprechenden Maßnahmen lediglich an die abweichende Abfrage- oder Skriptsprache des Datenbanksystems angepasst werden. Ein weiterer Grund dafür besteht darin, dass ein generalistischer Ansatz gewählt wurde, um ein Vorgehensmodell für alle Arten von nicht-relationalen Datenbanksystemen zu schaffen, anstatt sich z. B. auf die Absicherung von Schlüssel-Werte-Datenbanken zu beschränken.

3. In welcher Hinsicht weißt das Vorgehensmodell Defizite auf?

Das Vorgehensmodell sieht in seiner jetzigen Form keine Einstufung der Anforderungen hinsichtlich ihrer Kritikalität vor, sodass diese hinsichtlich ihrer Relevanz alle gleichermaßen gewichtet werden. Dies hat zur Folge, dass das eventuelle Fehlen von kritischen Sicherheitsanforderungen in der obigen Auswertung genauso stark ins Gewicht fällt wie Anforderungen, die als nebensächlich angesehen werden können. Dies hat somit Auswirkungen auf die Aussagekraft des Erfüllungsgrades, sodass in der jetzigen Entwicklungsstufe des Vorgehensmodells die Betrachtung der Auswertung nach Anforderungskategorien effektiver auf Defizite im unmittelbaren Vergleich mehrerer Datenbanksysteme hinweisen kann als die zusammengefasste Endauswertung.
Darüber hinaus wurden bei der Entwicklung des Vorgehensmodells nur die Standards berücksichtigt, die sich mit vollwertigen DBMS befassen. Diese Entscheidung wurde nicht vorsätzlich getroffen, sondern ergab sich vielmehr aus den für eine Analyse zur Verfügung stehenden CIS-Benchmarks und STIGs-Leitfäden für nicht-relationale Datenbanksysteme. Aufgrund der Ausschlussdefinition, die mit dem Begriff NoSQL verbunden ist, erstreckt sich der mögliche Betrachtungsrahmen jedoch auf eine Reihe weiterer, weniger „ausgereifter“ bzw. weniger komplexer Datenbanksysteme. Daher könnte es vor dem Hintergrund des verankerten Demingkreises sinnvoll sein, weitere nicht-relationale Datenbanksysteme zu analysieren, um weitere Anforderungen für z. B. In-Memory-Datenbanken wie Memcached zu identifizieren. Es ist jedoch anzumerken, dass es nahezu ausgeschlossen ist, alle existierenden Anforderungen abschließend zu identifizieren und in einem einzigen Vorgehensmodell zusammenzufassen, da zum einen stetig neue Datenbanksysteme hinzukommen und zum anderen der Umfang der Begriffsbestimmung zu umfangreich ist.

6 Zusammenfassung und Ausblick

Gegenstand dieser Master-Thesis war die Konzeption eines Vorgehensmodells zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme. Der Geltungsbereich wurde auf technische Maßnahmen beschränkt, während organisatorische und prozessuale Aspekte nicht berücksichtigt wurden.

Die Entwicklung erfolgte auf der Grundlage mehrerer etablierter Standards verschiedener Organisationen. Zum einen wurden alle relevanten prozess- und systemorientierten BSI IT-Grundschutz-Bausteine untersucht, woraus sich aufgrund ihres generischen Charakters ein sehr weit gefasster Maßnahmenkatalog ergab. Zum anderen wurden 3 CIS-Benchmarks und 2 STIGs-Leitfäden ausgewertet, die sich jeweils speziell mit der konkreten technischen Absicherung eines bestimmten nicht-relationalen Datenbanksystems befassen. Deren einzelne Maßnahmen wurden ebenfalls untersucht und diejenigen ausgeklammert, die im Rahmen des definierten Geltungsbereichs keine Verwendung finden konnten. Schließlich wurden alle aus diesem Prozess hervorgegangenen 282 Einzelmaßnahmen aus BSI, CIS und DISA STIGs zu insgesamt 82 Anforderungen zusammengefasst, die zusammen den Rahmen für das neue Vorgehensmodell bilden. Für jede dieser Anforderungen wurde eine detaillierte Beschreibung formuliert und, falls möglich, konkrete technische Umsetzungshinweise auf der Grundlage weiterer Best-Practice-Ansätze gegeben.

Im Anschluss erfolgte die Anwendung des Vorgehensmodells für die Community-Editionen der Datenbanksysteme Neo4j und Redis, in deren Verlauf die erforderlichen technischen Maßnahmen zur Realisierung einer gehärteten Konfiguration im Detail erarbeitet wurden. Für Redis wurden diese außerdem in ein InSpec-Compliance-Profil übertragen, durch dessen Ausführung ein automatisierter Soll-Ist-Abgleich der Konfiguration im Hinblick auf die Konformität vorgenommen werden kann.

Die abschließende Auswertung hinsichtlich des Erfüllungsgrades zeigte, dass gerundet für beide Datenbanksysteme nur etwa 65 % der Anforderungen umgesetzt werden konnten. Insbesondere bei der Authentifizierung (Punktwert-Abstand zu Neo4j in Höhe von $\approx 11,54$ %) und Autorisierung (Punktwert-Abstand zu Neo4j in Höhe von $20,00$ %) zeigten sich Unterschiede. In diesen Bereichen schnitt Redis signifikant

besser ab, was vor allem auf die überwiegend nicht vorhandene Konfigurierbarkeit beider Anforderungskategorien in der Community-Edition von Neo4j zurückzuführen ist. Demgegenüber konnte Neo4j mehr Anforderungen an den sicheren Betrieb (Punktwert-Abstand zu Redis in Höhe von $\approx 7,69$ %) erfüllen, da es im Gegensatz zu Redis über getrennte Schnittstellen für administrative und datenbezogene Zugriffe sowie über eine Funktionalität zur Vermeidung dynamischer Code-Ausführungen verfügt. Im Hinblick auf die analysierten Community-Editionen muss allerdings zur sachgerechten Einordnung der Untersuchungsergebnisse darauf hingewiesen werden, dass diese, im Gegensatz zu den erhältlichen Enterprise-Varianten, ein erhebliches Defizit hinsichtlich der verfügbaren Sicherheitsfunktionalitäten aufweisen. Eine Anwendung des Vorgehensmodells auf die Enterprise-Editionen würde aller Voraussicht nach zu einer deutlichen Steigerung des Erfüllungsgrades bei der Auswertung führen. Dies ist mutmaßlich auf eine gewollt starke Abgrenzung der beiden Varianten zurückzuführen, sodass die kostenfreien Community-Editionen nur in den wenigsten Fällen für den Einsatz in einer Produktivumgebung geeignet sind.

6.1 Einordnung des Vorgehensmodells

Mit dem Vorgehensmodell wurde ein iteratives Instrument zur Identifikation von technischen Härtungsmaßnahmen für Datenbanksysteme geschaffen. In seiner jetzigen Form mit 82 Anforderungen kann es sowohl auf relationale als auch auf nicht-relationale Datenbanksysteme angewendet werden. Dieser Umstand stellt jedoch kein qualitatives Defizit oder eine Verfehlung des Forschungsziels der Master-Thesis dar, da zur Entwicklung eines Vorgehensmodells für alle Arten von nicht-relationalen Datenbanksystemen ein generalistischer Ansatz gewählt werden muss. Vielmehr unterstreicht diese Tatsache somit die vorliegende und folglich auch zwingend notwendige Flexibilität, die vor dem Hintergrund der Definition des Begriffes NoSQL im Sinne von „Not only SQL“ unbedingt berücksichtigt werden muss. Denn auch ein nicht-relationales Datenbanksystem kann Funktionen von relationalen Datenbanken enthalten, soweit auch alternative Ansätze und Funktionalitäten, insbesondere zu SQL, zur Verfügung stehen.
Durch die Anwendung des Vorgehensmodells auf andere, weniger fortschrittliche bzw. weniger komplexe nicht-relationale Datenbanksysteme ist zu erwarten, dass einige Maßnahmen identifiziert werden können, die sich nicht eindeutig einer der bestehenden 82 Anforderungen zuordnen lassen. Diese können vor dem Hintergrund des in Anforderung A1 verankerten Demingkreises dazu genutzt werden, das Vorgehensmodell um weitere Anforderungen zu ergänzen und damit zu vergrößern. Auf

diese Weise kann auch eine weitere Präzisierung und damit eine klarere Abgrenzung zur Verwendung bei relationalen Datenbanksystemen erfolgen.

Ein verbleibendes Problem stellt die fehlende Einstufung der Anforderungen in Bezug auf ihre Kritikalität dar. Dadurch verliert der errechnete Erfüllungsgrad an Aussagekraft, sodass im Rahmen einer Fortschreibung ebenfalls eine entsprechende Bewertung eingeführt werden sollte. Als Vorbild kann dabei die Vorgehensweise von InSpec dienen, in der auf Basis des Common Vulnerability Scoring System (CVSS) ein numerischer Wert zwischen $0,0$ und $1,0$ zur Kennzeichnung der Kritikalität für jede Kontrolle spezifiziert werden kann [96].

Grundsätzlich bietet das Vorgehensmodell aber schon jetzt eine wertvolle Arbeitsgrundlage für die Ableitung von technischen Härtungsmaßnahmen, insbesondere für solche Systeme, für die bisher noch gar keine Leitfäden von etablierten Organisationen zur Verfügung stehen.

6.2 Schlusswort

Grundsätzlich muss für den produktiven Einsatz von nicht-relationalen Datenbanksystemen immer auch die Sicherheit berücksichtigt werden. Dabei ist zu beachten, dass die Community-Editionen im Gegensatz zu den entsprechenden Enterprise-Varianten häufig nicht über die für einen sicheren Betrieb erforderlichen Sicherheitsfunktionalitäten verfügen. Unabhängig davon wurde bei der Anwendung des Vorgehensmodells auch deutlich, dass nicht-relationale Datenbanksysteme oft keine sichere Standard-Konfiguration enthalten und selbst grundlegende Sicherheitsfunktionalitäten erst explizit aktiviert werden müssen. Dies gilt sowohl für die Community- als auch für die Enterprise-Editionen, sodass grundsätzlich eine Härtungsanalyse, etwa unter Zuhilfenahme des Vorgehensmodells, durchgeführt werden muss. Der Grund für diesen Umstand ist unter anderem darauf zurückzuführen, dass die Konfiguration bestimmter Sicherheitsfunktionen dem Anwendungszweck einer nicht-relationalen Datenbank zuwiderlaufen kann. Schließlich ist die große Anzahl an vorhandenen nicht-relationalen Datenbanksystemen nicht zuletzt auf die Vielzahl an technischen Problemstellungen zurückzuführen, die durch die Entwicklung des entsprechenden Systems gezielt adressiert werden soll. Das häufigste durch Sicherheitsmaßnahmen negativ beeinflusste Entwicklungsziel ist dabei die Performance. Anstatt jedoch alle Sicherheitsfunktionen standardmäßig zu deaktivieren, sollte ein ausgewogener Kompromiss angestrebt und umgesetzt werden, der möglichst sicher ist. Dieser Prozess kann durch das erarbeitete Vorgehensmodell gut begleitet werden.

A Hilfsmittel für die Verarbeitung der „Have I Been Pwned?“ API

Das nachstehend in Listing 9 gezeigte Python-Skript kann für den Zugriff auf die „Have I Been Pwned?“ API verwendet werden:

```
1  from datetime import datetime
2  import json
3  import locale
4  import requests
5
6
7  def get_hibp_breaches():
8      breaches =
       ↪  requests.get("https://haveibeenpwned.com/api/v3/breaches").json()
9      filtered_list = [x for x in breaches if
       ↪  x["BreachDate"].startswith("2022")]
10     return sorted(filtered_list, key=lambda x:
       ↪  datetime.strptime(x['BreachDate'], '%Y-%m-%d'))
11
12
13 def print_relational():
14     relational_list = get_hibp_breaches()
15     fk_dict = {b: a for a, b in enumerate(sorted(set([x['Title'] for x in
       ↪  relational_list])), 1)}
16
17     print("""\\begin{center}
18 \\begin{longtable}{|l|l|l|}
19 \\hline
20 \\rowcolor{gray!30!white}\\textbf{GID} & \\textbf{Name des Unternehmens} &
   ↪  \\textbf{Domain}\\\\
21 \\hline
22 \\endfirsthead""")
23     for k, v in fk_dict.items():
24         print(f"{v} & {k} & \\url{{{next(x['Domain'] for x in
           ↪  relational_list if x['Title'] == k)}}}\\\\\n\\hline")
25     print("""\\caption{Beispiel eines Relationsschemas "Geschädigter"}\\lab
       ↪  el{tab:beispiel-eines-relationsschemas-geschaedigter}
```

```
26 \\end{longtable}
27 \\end{center}""")
28     print("""\\begin{center}
29 \\begin{longtable}{|l|l|p{2.5cm}|p{2.5cm}|p{5.9cm}|}
30 \\hline
31 \\rowcolor{gray!30!white}\\textbf{DID} & \\textbf{GID} & \\textbf{Datum der
   ↪  Sicherheitsverletzung} & \\textbf{Anzahl betroffener Konten} &
   ↪  \\textbf{Art der kompromittierten Daten}\\\\
32 \\hline
33 \\endfirsthead""")
34     for i, x in enumerate(relational_list, start=1):
35         print(f"{i} & {fk_dict[x['Title']]} & {x['BreachDate']} &
           ↪  {x['PwnCount']:n} & "
36               f"{', '.join(x['DataClasses'])}\\\\\n\\hline")
37     print("""\\caption{Beispiel eines Relationsschemas "Kompromittierte
       ↪  Datensätze"}\\label{tab:beispiel-eines-relationsschemas-kompromitti
       ↪  erte-datensaetze}
38 \\end{longtable}
39 \\end{center}""")
40
41
42 def print_document():
43     global document_list
44     document_list = get_hibp_breaches()
45
46     for i, x in enumerate(document_list):
47         document_list[i]['Name des Unternehmens'] =
           ↪  document_list[i].pop('Title')
48         document_list[i]['Domain'] = document_list[i].pop('Domain')
49         document_list[i]['Datum der Sicherheitsverletzung'] =
           ↪  document_list[i].pop('BreachDate')
50         document_list[i]['Anzahl betroffener Konten'] =
           ↪  document_list[i].pop('PwnCount')
51         document_list[i]['Art der kompromittierten Daten'] =
           ↪  document_list[i].pop('DataClasses')
52         for y in ['Name', 'AddedDate', 'ModifiedDate', 'Description',
           ↪  'LogoPath', 'IsVerified', 'IsFabricated',
53                   'IsSensitive', 'IsRetired', 'IsSpamList', 'IsMalware']:
54             document_list[i].pop(y)
55     print(json.dumps(document_list, indent=4))
56
57
58 def print_redis():
59     for i, x in enumerate(document_list, start=1):
```

```
60          for k, v in x.items():
61              if isinstance(v, list):
62                  [print(f"SADD {k.lower().replace(' ', '-')}#{i} \"{y}\"")
                    ↪  for y in v]
63              else:
64                  print(f"SET {k.lower().replace(' ', '-')}#{i} \"{v}\"")
65          print()
66
67
68  def print_column():
69      print("""\\begin{center}
70  \\begin{longtable}{|p{2.5cm}|p{0.1cm}|p{4.5cm}|p{0.1cm}|p{2.5cm}|p{0.1cm}|p
    ↪  {2.3cm}|p{0.1cm}|p{7.3cm}|}
71  \\cline{1-1}\\cline{3-3}\\cline{5-5}\\cline{7-7}\\cline{9-9}
72  \\cellcolor{YellowGreen}\\textbf{Name des Unternehmens} &&
    ↪  \\cellcolor{GreenYellow}\\textbf{Domain} &&
    ↪  \\cellcolor{Salmon}\\textbf{Datum der Sicherheitsverletzung} &&
    ↪  \\cellcolor{YellowOrange}\\textbf{Anzahl betroffener Konten} &&
    ↪  \\cellcolor{SkyBlue}\\textbf{Art der kompromittierten Daten}\\\\
73  \\cline{1-1}\\cline{3-3}\\cline{5-5}\\cline{7-7}\\cline{9-9}
74  \\endfirsthead""")
75      for x in document_list:
76          print(
77              f"{x['Name des Unternehmens']} && \\url{{{x['Domain']}}} &&
                ↪  {x['Datum der Sicherheitsverletzung']} && {x['Anzahl
                ↪  betroffener Konten']:n} && {', '.join(x['Art der
                ↪  kompromittierten Daten'])}\\\\\n\\cline{{1-1}}\\cline{{3-3}
                ↪  }\\cline{{5-5}}\\cline{{7-7}}\\cline{{9-9}}")
78      print("""\\caption{Aufbereitete Ausgabe aus der "Have I Been Pwned?"
        ↪  API als Beispiel für spaltenorientierte
        ↪  Datenbanken}\\label{tab:aufbereitete-ausgabe-aus-der-have-i-been-pw
        ↪  ned-api-als-beispiel-fuer-spaltenorientierte-datenbanken}
79  \\end{longtable}
80  \\end{center}""")
81      print("""\\begin{center}
82  \\begin{longtable}[r]{|p{3.1cm}|p{0.1cm}|p{4.4cm}|p{0.1cm}|p{2.5cm}|p{0.1cm
    ↪  }|p{0.5cm}|}
83  \\cline{1-1}\\cline{3-3}\\cline{5-5}\\cline{7-7}
84  \\cellcolor{YellowGreen}\\textbf{Name des Unternehmens} &&
    ↪  \\cellcolor{GreenYellow}\\textbf{Domain} &&
    ↪  \\cellcolor{Salmon}\\textbf{Datum der Sicherheitsverletzung} &&
    ↪  \\cellcolor{gray!30!white}\\textbf{...}\\\\
85  \\cline{1-1}\\cline{3-3}\\cline{5-5}\\cline{7-7}
86  \\endfirsthead""")
```

```
87      for x in document_list:
88          print(
89              f"{x['Name des Unternehmens']} && \\url{{{x['Domain']}}} &&
                ↪  {x['Datum der Sicherheitsverletzung']} && ...\\\\\n\\cline{
                ↪  {1-1}}\\cline{{3-3}}\\cline{{5-5}}\\cline{{7-7}}")
90      print("""\\caption{Gekürzte aufbereitete Ausgabe aus der "Have I Been
        ↪  Pwned?" API als Beispiel für spaltenorientierte
        ↪  Datenbanken}\\label{tab:gekuerzte-aufbereitete-ausgabe-aus-der-have
        ↪  -i-been-pwned-api-als-beispiel-fuer-spaltenorientierte-datenbanken}
91  \\end{longtable}
92  \\end{center}""")
93
94
95  if __name__ == '__main__':
96      locale.setlocale(locale.LC_ALL, 'de_DE.UTF-8')
97      print_relational()
98      print_document()
99      print_redis()
100     print_column()
```

Listing 9: Hilfsmittel für die Verarbeitung der „Have I Been Pwned?“ API

B „Have I Been Pwned?“ Ausgabe für relationale Datenbanken

Die folgende Tabelle 18 zeigt die ungekürzte Ausgabe eines Relationsschemas „Geschädigter“ aus der „Have I Been Pwned?“ API als Beispiel für relationale Datenbanken:

GID	Name des Unternehmens	Domain
1	Amart Furniture	`amartfurniture.com.au`
2	BlackBerry Fans	`blackberryfans.org`
3	CDEK	`cdek.ru`
4	Doxbin	`doxbin.com`
5	Fanpass	`fanpass.co.uk`
6	GiveSendGo	`givesendgo.com`
7	La Poste Mobile	`lapostemobile.fr`
8	MacGeneration	`macg.co`
9	Mangatoon	`mangatoon.mobi`
10	NVIDIA	`nvidia.com`
11	PayHere	`payhere.lk`
12	QuestionPro	`questionpro.com`
13	Shitexpress	`shitexpress.com`
14	Twitter	`twitter.com`

Tabelle 18: Aufbereitete Ausgabe eines Relationsschemas „Geschädigter“ aus der „Have I Been Pwned?“ API als Beispiel für relationale Datenbanken

Die folgende Tabelle 19 zeigt die ungekürzte Ausgabe eines Relationsschemas „Kompromittierte Datensätze“ aus der „Have I Been Pwned?“ API als Beispiel für relationale Datenbanken:

DID	**GID**	**Datum der Sicherheitsverletzung**	**Anzahl betroffener Konten**	**Art der kompromittierten Daten**
1	14	2022-01-01	6.682.453	Bios, Email addresses, Geographic locations, Names, Phone numbers, Profile photos, Usernames
2	4	2022-01-05	370.794	Browser user agent details, Email addresses, Passwords, Usernames
3	8	2022-01-29	101.004	Email addresses, Passwords, Usernames
4	6	2022-02-07	89.966	Email addresses, Geographic locations, Names, Purchases
5	10	2022-02-23	71.335	Email addresses, Passwords
6	3	2022-03-09	19.218.203	Email addresses, Names, Phone numbers
7	11	2022-03-27	1.580.249	Email addresses, IP addresses, Names, Partial credit card data, Phone numbers, Physical addresses, Purchases
8	5	2022-04-30	112.251	Email addresses, Genders, Names, Partial dates of birth, Passwords, Phone numbers, Physical addresses, Purchases, Social media profiles
9	2	2022-05-06	174.168	Email addresses, IP addresses, Passwords, Usernames
10	9	2022-05-13	23.040.238	Auth tokens, Avatars, Email addresses, Genders, Names, Passwords, Social media profiles, Usernames

11	1	2022-05-16	108.940	Email addresses, Names, Passwords, Phone numbers, Physical addresses
12	12	2022-05-21	22.229.637	Browser user agent details, Email addresses, IP addresses, Survey results
13	7	2022-07-04	533.886	Bank account numbers, Dates of birth, Email addresses, Genders, Names, Phone numbers, Physical addresses
14	13	2022-08-08	23.817	Email addresses, IP addresses, Names, Physical addresses, Private messages, Purchases

Tabelle 19: Aufbereitete Ausgabe eines Relationsschemas „Kompromittierte Datensätze“ aus der „Have I Been Pwned?“ API als Beispiel für relationale Datenbanken

C „Have I Been Pwned?“ Ausgabe für dokumentenorientierte Datenbanken

Das folgende Listing 10 zeigt die ungekürzte Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für dokumentenorientierte Datenbanken:


```
1  [
2      {
3          "Name des Unternehmens": "Twitter",
4          "Domain": "twitter.com",
5          "Datum der Sicherheitsverletzung": "2022-01-01",
6          "Anzahl betroffener Konten": 6682453,
7          "Art der kompromittierten Daten": [
8              "Bios",
9              "Email addresses",
10             "Geographic locations",
11             "Names",
12             "Phone numbers",
13             "Profile photos",
14             "Usernames"
15         ]
16     },
17     {
18         "Name des Unternehmens": "Doxbin",
19         "Domain": "doxbin.com",
20         "Datum der Sicherheitsverletzung": "2022-01-05",
21         "Anzahl betroffener Konten": 370794,
22         "Art der kompromittierten Daten": [
23             "Browser user agent details",
24             "Email addresses",
25             "Passwords",
26             "Usernames"
27         ]
28     },
29     {
30         "Name des Unternehmens": "MacGeneration",
31         "Domain": "macg.co",
```

```
32          "Datum der Sicherheitsverletzung": "2022-01-29",
33          "Anzahl betroffener Konten": 101004,
34          "Art der kompromittierten Daten": [
35              "Email addresses",
36              "Passwords",
37              "Usernames"
38          ]
39      },
40      {
41          "Name des Unternehmens": "GiveSendGo",
42          "Domain": "givesendgo.com",
43          "Datum der Sicherheitsverletzung": "2022-02-07",
44          "Anzahl betroffener Konten": 89966,
45          "Art der kompromittierten Daten": [
46              "Email addresses",
47              "Geographic locations",
48              "Names",
49              "Purchases"
50          ]
51      },
52      {
53          "Name des Unternehmens": "NVIDIA",
54          "Domain": "nvidia.com",
55          "Datum der Sicherheitsverletzung": "2022-02-23",
56          "Anzahl betroffener Konten": 71335,
57          "Art der kompromittierten Daten": [
58              "Email addresses",
59              "Passwords"
60          ]
61      },
62      {
63          "Name des Unternehmens": "CDEK",
64          "Domain": "cdek.ru",
65          "Datum der Sicherheitsverletzung": "2022-03-09",
66          "Anzahl betroffener Konten": 19218203,
67          "Art der kompromittierten Daten": [
68              "Email addresses",
69              "Names",
70              "Phone numbers"
71          ]
72      },
73      {
74          "Name des Unternehmens": "PayHere",
75          "Domain": "payhere.lk",
```

```
76          "Datum der Sicherheitsverletzung": "2022-03-27",
77          "Anzahl betroffener Konten": 1580249,
78          "Art der kompromittierten Daten": [
79              "Email addresses",
80              "IP addresses",
81              "Names",
82              "Partial credit card data",
83              "Phone numbers",
84              "Physical addresses",
85              "Purchases"
86          ]
87      },
88      {
89          "Name des Unternehmens": "Fanpass",
90          "Domain": "fanpass.co.uk",
91          "Datum der Sicherheitsverletzung": "2022-04-30",
92          "Anzahl betroffener Konten": 112251,
93          "Art der kompromittierten Daten": [
94              "Email addresses",
95              "Genders",
96              "Names",
97              "Partial dates of birth",
98              "Passwords",
99              "Phone numbers",
100             "Physical addresses",
101             "Purchases",
102             "Social media profiles"
103         ]
104     },
105     {
106         "Name des Unternehmens": "BlackBerry Fans",
107         "Domain": "blackberryfans.org",
108         "Datum der Sicherheitsverletzung": "2022-05-06",
109         "Anzahl betroffener Konten": 174168,
110         "Art der kompromittierten Daten": [
111             "Email addresses",
112             "IP addresses",
113             "Passwords",
114             "Usernames"
115         ]
116     },
117     {
118         "Name des Unternehmens": "Mangatoon",
119         "Domain": "mangatoon.mobi",
```

```
120         "Datum der Sicherheitsverletzung": "2022-05-13",
121         "Anzahl betroffener Konten": 23040238,
122         "Art der kompromittierten Daten": [
123             "Auth tokens",
124             "Avatars",
125             "Email addresses",
126             "Genders",
127             "Names",
128             "Passwords",
129             "Social media profiles",
130             "Usernames"
131         ]
132     },
133     {
134         "Name des Unternehmens": "Amart Furniture",
135         "Domain": "amartfurniture.com.au",
136         "Datum der Sicherheitsverletzung": "2022-05-16",
137         "Anzahl betroffener Konten": 108940,
138         "Art der kompromittierten Daten": [
139             "Email addresses",
140             "Names",
141             "Passwords",
142             "Phone numbers",
143             "Physical addresses"
144         ]
145     },
146     {
147         "Name des Unternehmens": "QuestionPro",
148         "Domain": "questionpro.com",
149         "Datum der Sicherheitsverletzung": "2022-05-21",
150         "Anzahl betroffener Konten": 22229637,
151         "Art der kompromittierten Daten": [
152             "Browser user agent details",
153             "Email addresses",
154             "IP addresses",
155             "Survey results"
156         ]
157     },
158     {
159         "Name des Unternehmens": "La Poste Mobile",
160         "Domain": "lapostemobile.fr",
161         "Datum der Sicherheitsverletzung": "2022-07-04",
162         "Anzahl betroffener Konten": 533886,
163         "Art der kompromittierten Daten": [
```

```
164                 "Bank account numbers",
165                 "Dates of birth",
166                 "Email addresses",
167                 "Genders",
168                 "Names",
169                 "Phone numbers",
170                 "Physical addresses"
171             ]
172         },
173         {
174             "Name des Unternehmens": "Shitexpress",
175             "Domain": "shitexpress.com",
176             "Datum der Sicherheitsverletzung": "2022-08-08",
177             "Anzahl betroffener Konten": 23817,
178             "Art der kompromittierten Daten": [
179                 "Email addresses",
180                 "IP addresses",
181                 "Names",
182                 "Physical addresses",
183                 "Private messages",
184                 "Purchases"
185             ]
186         }
187 ]
```

Listing 10: Aufbereitete Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für dokumentenorientierte Datenbanken

D „Have I Been Pwned?“ Ausgabe für Schlüssel-Werte Datenbanken

Das folgende Listing 11 zeigt die ungekürzte Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für Schlüssel-Werte-Datenbanken:

```
SET name-des-unternehmens#1 "Twitter"
SET domain#1 "twitter.com"
SET datum-der-sicherheitsverletzung#1 "2022-01-01"
SET anzahl-betroffener-konten#1 "6682453"
SADD art-der-kompromittierten-daten#1 "Bios"
SADD art-der-kompromittierten-daten#1 "Email addresses"
SADD art-der-kompromittierten-daten#1 "Geographic locations"
SADD art-der-kompromittierten-daten#1 "Names"
SADD art-der-kompromittierten-daten#1 "Phone numbers"
SADD art-der-kompromittierten-daten#1 "Profile photos"
SADD art-der-kompromittierten-daten#1 "Usernames"

SET name-des-unternehmens#2 "Doxbin"
SET domain#2 "doxbin.com"
SET datum-der-sicherheitsverletzung#2 "2022-01-05"
SET anzahl-betroffener-konten#2 "370794"
SADD art-der-kompromittierten-daten#2 "Browser user agent details"
SADD art-der-kompromittierten-daten#2 "Email addresses"
SADD art-der-kompromittierten-daten#2 "Passwords"
SADD art-der-kompromittierten-daten#2 "Usernames"

SET name-des-unternehmens#3 "MacGeneration"
SET domain#3 "macg.co"
SET datum-der-sicherheitsverletzung#3 "2022-01-29"
SET anzahl-betroffener-konten#3 "101004"
SADD art-der-kompromittierten-daten#3 "Email addresses"
SADD art-der-kompromittierten-daten#3 "Passwords"
SADD art-der-kompromittierten-daten#3 "Usernames"

SET name-des-unternehmens#4 "GiveSendGo"
SET domain#4 "givesendgo.com"
```

```
32 SET datum-der-sicherheitsverletzung#4 "2022-02-07"
33 SET anzahl-betroffener-konten#4 "89966"
34 SADD art-der-kompromittierten-daten#4 "Email addresses"
35 SADD art-der-kompromittierten-daten#4 "Geographic locations"
36 SADD art-der-kompromittierten-daten#4 "Names"
37 SADD art-der-kompromittierten-daten#4 "Purchases"
38
39 SET name-des-unternehmens#5 "NVIDIA"
40 SET domain#5 "nvidia.com"
41 SET datum-der-sicherheitsverletzung#5 "2022-02-23"
42 SET anzahl-betroffener-konten#5 "71335"
43 SADD art-der-kompromittierten-daten#5 "Email addresses"
44 SADD art-der-kompromittierten-daten#5 "Passwords"
45
46 SET name-des-unternehmens#6 "CDEK"
47 SET domain#6 "cdek.ru"
48 SET datum-der-sicherheitsverletzung#6 "2022-03-09"
49 SET anzahl-betroffener-konten#6 "19218203"
50 SADD art-der-kompromittierten-daten#6 "Email addresses"
51 SADD art-der-kompromittierten-daten#6 "Names"
52 SADD art-der-kompromittierten-daten#6 "Phone numbers"
53
54 SET name-des-unternehmens#7 "PayHere"
55 SET domain#7 "payhere.lk"
56 SET datum-der-sicherheitsverletzung#7 "2022-03-27"
57 SET anzahl-betroffener-konten#7 "1580249"
58 SADD art-der-kompromittierten-daten#7 "Email addresses"
59 SADD art-der-kompromittierten-daten#7 "IP addresses"
60 SADD art-der-kompromittierten-daten#7 "Names"
61 SADD art-der-kompromittierten-daten#7 "Partial credit card data"
62 SADD art-der-kompromittierten-daten#7 "Phone numbers"
63 SADD art-der-kompromittierten-daten#7 "Physical addresses"
64 SADD art-der-kompromittierten-daten#7 "Purchases"
65
66 SET name-des-unternehmens#8 "Fanpass"
67 SET domain#8 "fanpass.co.uk"
68 SET datum-der-sicherheitsverletzung#8 "2022-04-30"
69 SET anzahl-betroffener-konten#8 "112251"
70 SADD art-der-kompromittierten-daten#8 "Email addresses"
71 SADD art-der-kompromittierten-daten#8 "Genders"
72 SADD art-der-kompromittierten-daten#8 "Names"
73 SADD art-der-kompromittierten-daten#8 "Partial dates of birth"
74 SADD art-der-kompromittierten-daten#8 "Passwords"
75 SADD art-der-kompromittierten-daten#8 "Phone numbers"
```

```
76 SADD art-der-kompromittierten-daten#8 "Physical addresses"
77 SADD art-der-kompromittierten-daten#8 "Purchases"
78 SADD art-der-kompromittierten-daten#8 "Social media profiles"
79
80 SET name-des-unternehmens#9 "BlackBerry Fans"
81 SET domain#9 "blackberryfans.org"
82 SET datum-der-sicherheitsverletzung#9 "2022-05-06"
83 SET anzahl-betroffener-konten#9 "174168"
84 SADD art-der-kompromittierten-daten#9 "Email addresses"
85 SADD art-der-kompromittierten-daten#9 "IP addresses"
86 SADD art-der-kompromittierten-daten#9 "Passwords"
87 SADD art-der-kompromittierten-daten#9 "Usernames"
88
89 SET name-des-unternehmens#10 "Mangatoon"
90 SET domain#10 "mangatoon.mobi"
91 SET datum-der-sicherheitsverletzung#10 "2022-05-13"
92 SET anzahl-betroffener-konten#10 "23040238"
93 SADD art-der-kompromittierten-daten#10 "Auth tokens"
94 SADD art-der-kompromittierten-daten#10 "Avatars"
95 SADD art-der-kompromittierten-daten#10 "Email addresses"
96 SADD art-der-kompromittierten-daten#10 "Genders"
97 SADD art-der-kompromittierten-daten#10 "Names"
98 SADD art-der-kompromittierten-daten#10 "Passwords"
99 SADD art-der-kompromittierten-daten#10 "Social media profiles"
100 SADD art-der-kompromittierten-daten#10 "Usernames"
101
102 SET name-des-unternehmens#11 "Amart Furniture"
103 SET domain#11 "amartfurniture.com.au"
104 SET datum-der-sicherheitsverletzung#11 "2022-05-16"
105 SET anzahl-betroffener-konten#11 "108940"
106 SADD art-der-kompromittierten-daten#11 "Email addresses"
107 SADD art-der-kompromittierten-daten#11 "Names"
108 SADD art-der-kompromittierten-daten#11 "Passwords"
109 SADD art-der-kompromittierten-daten#11 "Phone numbers"
110 SADD art-der-kompromittierten-daten#11 "Physical addresses"
111
112 SET name-des-unternehmens#12 "QuestionPro"
113 SET domain#12 "questionpro.com"
114 SET datum-der-sicherheitsverletzung#12 "2022-05-21"
115 SET anzahl-betroffener-konten#12 "22229637"
116 SADD art-der-kompromittierten-daten#12 "Browser user agent details"
117 SADD art-der-kompromittierten-daten#12 "Email addresses"
118 SADD art-der-kompromittierten-daten#12 "IP addresses"
119 SADD art-der-kompromittierten-daten#12 "Survey results"
```

```
120
121 SET name-des-unternehmens#13 "La Poste Mobile"
122 SET domain#13 "lapostemobile.fr"
123 SET datum-der-sicherheitsverletzung#13 "2022-07-04"
124 SET anzahl-betroffener-konten#13 "533886"
125 SADD art-der-kompromittierten-daten#13 "Bank account numbers"
126 SADD art-der-kompromittierten-daten#13 "Dates of birth"
127 SADD art-der-kompromittierten-daten#13 "Email addresses"
128 SADD art-der-kompromittierten-daten#13 "Genders"
129 SADD art-der-kompromittierten-daten#13 "Names"
130 SADD art-der-kompromittierten-daten#13 "Phone numbers"
131 SADD art-der-kompromittierten-daten#13 "Physical addresses"
132
133 SET name-des-unternehmens#14 "Shitexpress"
134 SET domain#14 "shitexpress.com"
135 SET datum-der-sicherheitsverletzung#14 "2022-08-08"
136 SET anzahl-betroffener-konten#14 "23817"
137 SADD art-der-kompromittierten-daten#14 "Email addresses"
138 SADD art-der-kompromittierten-daten#14 "IP addresses"
139 SADD art-der-kompromittierten-daten#14 "Names"
140 SADD art-der-kompromittierten-daten#14 "Physical addresses"
141 SADD art-der-kompromittierten-daten#14 "Private messages"
142 SADD art-der-kompromittierten-daten#14 "Purchases"
```

Listing 11: Aufbereitete Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für Schlüssel-Werte-Datenbanken

E „Have I Been Pwned?“ Ausgabe für spaltenorientierte Datenbanken

Die folgende Tabelle 20 zeigt die ungekürzte Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für spaltenorientierte Datenbanken:

Name des Unternehmens	Domain	Datum der Sicherheitsverletzung	Anzahl betroffener Konten	Art der kompromittierten Daten
Twitter	`twitter.com`	2022-01-01	6.682.453	Bios, Email addresses, Geographic locations, Names, Phone numbers, Profile photos, Usernames
Doxbin	`doxbin.com`	2022-01-05	370.794	Browser user agent details, Email addresses, Passwords, Usernames
MacGeneration	`macg.co`	2022-01-29	101.004	Email addresses, Passwords, Usernames
GiveSendGo	`givesendgo.com`	2022-02-07	89.966	Email addresses, Geographic locations, Names, Purchases
NVIDIA	`nvidia.com`	2022-02-23	71.335	Email addresses, Passwords

CDEK	cdek.ru	2022-03-09	19.218.203	Email addresses, Names, Phone numbers
PayHere	payhere.lk	2022-03-27	1.580.249	Email addresses, IP addresses, Names, Partial credit card data, Phone numbers, Physical addresses, Purchases
Fanpass	fanpass.co.uk	2022-04-30	112.251	Email addresses, Genders, Names, Partial dates of birth, Passwords, Phone numbers, Physical addresses, Purchases, Social media profiles
BlackBerry Fans	blackberryfans.org	2022-05-06	174.168	Email addresses, IP addresses, Passwords, Usernames
Mangatoon	mangatoon.mobi	2022-05-13	23.040.238	Auth tokens, Avatars, Email addresses, Genders, Names, Passwords, Social media profiles, Usernames
Amart Furniture	amartfurniture.com.au	2022-05-16	108.940	Email addresses, Names, Passwords, Phone numbers, Physical addresses
QuestionPro	questionpro.com	2022-05-21	22.229.637	Browser user agent details, Email addresses, IP addresses, Survey results
La Poste Mobile	lapostemobile.fr	2022-07-04	533.886	Bank account numbers, Dates of birth, Email addresses, Genders, Names, Phone numbers, Physical addresses
Shitexpress	shitexpress.com	2022-08-08	23.817	Email addresses, IP addresses, Names, Physical addresses, Private messages, Purchases

Brand New Tube	`brandnewtube.com`	2022-08-14	349.627	Email addresses, Genders, IP addresses, Passwords, Private messages, Usernames

Tabelle 20: Aufbereitete Ausgabe aus der „Have I Been Pwned?“ API als Beispiel für spaltenorientierte Datenbanken

F Identifikation aller Einzelanforderungen der IT-Grundschutz-Bausteine

Nachfolgend werden alle anwendbaren Maßnahmen der BSI IT-Grundschutz-Bausteine [78] mit einer fortlaufenden Nummer mit dem Präfix „B“ versehen, um eine Gruppierung und Referenzierung zu ermöglichen. Alle Maßnahmen, die nicht berücksichtigt werden, enthalten eine entsprechende Begründung.

ORP.4 Identitäts- und Berechtigungsmanagement

ORP.4.A1 Regelung für die Einrichtung und Löschung von Benutzern und Benutzergruppen [IT-Betrieb] (B)

× **„Es MUSS geregelt werden, wie Benutzerkennungen und Benutzergruppen einzurichten und zu löschen sind.“**
Entfällt, da die Verwaltung von Benutzerkennungen und Benutzergruppen nicht über eine technische Konfigurationsanpassung abgebildet werden kann, sondern durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Jede Benutzerkennung MUSS eindeutig einem Benutzer zugeordnet werden können.“**
Entfällt, da die Zuordnung von Benutzerkennungen durch ein externes IT-System, wie z. B. ein IAM- oder Identity Management (IdM)-System, oder alternativ durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B1 **„Benutzerkennungen, die längere Zeit inaktiv sind, SOLLTEN deaktiviert werden.“**

B2 **„Alle Benutzer und Benutzergruppen DÜRFEN NUR über separate administrative Rollen eingerichtet und gelöscht werden.“**

B3 **„Nicht benötigte Benutzerkennungen, wie z. B. standardmäßig eingerichtete Gastkonten oder Standard-Administratorkennungen, MÜSSEN geeignet deaktiviert oder gelöscht werden.“**

ORP.4.A2 Einrichtung, Änderung und Entzug von Berechtigungen [IT-Betrieb] (B)

B4 **„Benutzerkennungen und Berechtigungen DÜRFEN NUR aufgrund des tatsächlichen Bedarfs und der Notwendigkeit zur Aufgabenerfüllung vergeben werden (Prinzip der geringsten Berechtigungen, engl. Least Privileges und Erforderlichkeitsprinzip, engl. Need-to-know).“**

× **„Bei personellen Veränderungen MÜSSEN die nicht mehr benötigten Benutzerkennungen und Berechtigungen entfernt werden.“**
Entfällt, da die Berücksichtigung von personellen Veränderungen nicht über eine technische Konfigurationsanpassung abgebildet werden kann, sondern durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Beantragen Mitarbeiter Berechtigungen, die über den Standard hinausgehen, DÜRFEN diese NUR nach zusätzlicher Begründung und Prüfung vergeben werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B5 **„Zugriffsberechtigungen auf Systemverzeichnisse und -dateien SOLLTEN restriktiv eingeschränkt werden.“**

B6 **„Alle Berechtigungen MÜSSEN über separate administrative Rollen eingerichtet werden.“**

ORP.4.A8 Regelung des Passwortgebrauchs [Benutzer, IT-Betrieb] (B)

× **„Die Institution MUSS den Passwortgebrauch verbindlich regeln.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss. Die Mindestanforderungen an die Passwortkomplexität müssen jedoch als Teil der Konfiguration festgelegt werden, siehe Anforderung B9.

B7 **„Dabei MUSS geprüft werden, ob Passwörter als alleiniges Authentisierungsverfahren eingesetzt werden sollen, oder ob andere Authentisierungsmerkmale bzw. -verfahren zusätzlich zu oder anstelle von Passwörtern verwendet werden können.“**

B8 **„Passwörter DÜRFEN NICHT mehrfach verwendet werden.“**

× **„Für jedes IT-System bzw. jede Anwendung MUSS ein eigenständiges Passwort verwendet werden.“**
Entfällt, da eine anwendungsübergreifende Prüfung auf Wiederverwendung von Zugangsdaten außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

B9 **„Passwörter, die leicht zu erraten sind oder in gängigen Passwortlisten geführt werden, DÜRFEN NICHT verwendet werden.“**

× **„Passwörter MÜSSEN geheim gehalten werden. Sie DÜRFEN NUR dem Benutzer persönlich bekannt sein.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Passwörter DÜRFEN NUR unbeobachtet eingegeben werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Passwörter DÜRFEN NICHT auf programmierbaren Funktionstasten von Tastaturen oder Mäusen gespeichert werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Ein Passwort DARF NUR für eine Hinterlegung für einen Notfall schriftlich fixiert werden. Es MUSS dann sicher aufbewahrt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Die Nutzung eines Passwort-Managers SOLLTE geprüft werden. Bei Passwort-Managern mit Funktionen oder Plug-ins, mit denen Passwörter über Onlinedienste Dritter synchronisiert oder anderweitig an Dritte übertragen werden, MÜSSEN diese Funktionen und Plug-ins deaktiviert werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Ein Passwort MUSS gewechselt werden, wenn es unautorisierten Personen bekannt geworden ist oder der Verdacht dazu besteht.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss. Es müssen weitere technische Systeme zur Erkennung von Anzeichen einer Kompromittierung bereitgestellt werden, z.B. in Form eines SIEM-Systems. Durch die Erfassung aller sicherheitsrelevanten Ereignisse in Anforderung B61 werden die Voraussetzungen hierfür geschaffen.

ORP.4.A9 Identifikation und Authentisierung [IT-Betrieb] (B)

B10 **„Der Zugriff auf alle IT-Systeme und Dienste MUSS durch eine angemessene Identifikation und Authentisierung der zugreifenden Benutzer, Dienste oder IT-Systeme abgesichert sein.“**

B11 **„Vorkonfigurierte Authentisierungsmittel MÜSSEN vor dem produktiven Einsatz geändert werden.“**

ORP.4.A22 Regelung zur Passwortqualität [IT-Betrieb] (B)

B12 **„In Abhängigkeit von Einsatzzweck und Schutzbedarf MÜSSEN sichere Passwörter geeigneter Qualität gewählt werden.“**

B13 **„Das Passwort MUSS so komplex sein, dass es nicht leicht zu erraten ist.“**

× **„Das Passwort DARF NICHT zu kompliziert sein, damit der Benutzer in der Lage ist, das Passwort mit vertretbarem Aufwand regelmäßig zu verwenden.“**
Entfällt, da dies der Anforderung B12 widerspricht und zudem der Fokus dieser Master-Thesis auf einer optimalen technischen Absicherung liegt.

ORP.4.A23 Regelung für Passwort-verarbeitende Anwendungen und IT-Systeme [IT-Betrieb] (B)

B14 **„IT-Systeme oder Anwendungen SOLLTEN NUR mit einem validen Grund zum Wechsel des Passworts auffordern. Reine zeitgesteuerte Wechsel SOLLTEN vermieden werden.“**

B15 **„Es MÜSSEN Maßnahmen ergriffen werden, um die Kompromittierung von Passwörtern zu erkennen.“**

× **„Ist dies nicht möglich, so SOLLTE geprüft werden, ob die Nachteile eines zeitgesteuerten Passwortwechsels in Kauf genommen werden können und Passwörter in gewissen Abständen gewechselt werden.“**
Entfällt, da dies der Anforderung B14 widerspricht und zudem der Fokus dieser Master-Thesis auf einer optimalen technischen Absicherung liegt.

B16 **„Standardpasswörter MÜSSEN durch ausreichend starke Passwörter ersetzt und vordefinierte Kennungen MÜSSEN geändert werden.“**

× **„Es SOLLTE sichergestellt werden, dass die mögliche Passwortlänge auch im vollen Umfang von verarbeitenden IT-Systemen geprüft wird.“**
Entfällt, da eine anwendungsübergreifende Prüfung der Passwortlänge außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

B17 **„Nach einem Passwortwechsel DÜRFEN alte Passwörter NICHT mehr genutzt werden.“**

B18 **„Passwörter MÜSSEN so sicher wie möglich gespeichert werden.“**

× **„Bei Kennungen für technische Benutzer, Dienstkonten, Schnittstellen oder Vergleichbares SOLLTE ein Passwortwechsel sorgfältig geplant und gegebenenfalls mit den Anwendungsverantwortlichen abgestimmt werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B19 **„Bei der Authentisierung in vernetzten Systemen DÜRFEN Passwörter NICHT unverschlüsselt über unsichere Netze übertragen werden.“**

B20 **„Wenn Passwörter in einem Intranet übertragen werden, SOLLTEN sie verschlüsselt werden.“**

B21 **„Bei erfolglosen Anmeldeversuchen SOLLTE das System keinen Hinweis darauf geben, ob Passwort oder Benutzerkennung falsch sind.“**

ORP.4.A10 Schutz von Benutzerkennungen mit weitreichenden Berechtigungen [IT-Betrieb] (S)

B22 **„Benutzerkennungen mit weitreichenden Berechtigungen SOLLTEN mit einer Mehr-Faktor-Authentisierung, z. B. mit kryptografischen Zertifikaten, Chipkarten oder Token, geschützt werden.“**

ORP.4.A11 Zurücksetzen von Passwörtern [IT-Betrieb] (S)

B23 **„Für das Zurücksetzen von Passwörtern SOLLTE ein angemessenes sicheres Verfahren definiert und umgesetzt werden.“**

× **„Die Support-Mitarbeiter, die Passwörter zurücksetzen können, SOLLTEN entsprechend geschult werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Bei höherem Schutzbedarf des Passwortes SOLLTE eine Strategie definiert werden, falls ein Support-Mitarbeiter aufgrund fehlender sicherer Möglichkeiten der Übermittlung des Passwortes die Verantwortung nicht übernehmen kann.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

ORP.4.A12 Entwicklung eines Authentisierungskonzeptes für IT-Systeme und Anwendungen [IT-Betrieb] (S)

× **„Es SOLLTE ein Authentisierungskonzept erstellt werden. Darin SOLLTE für jedes IT-System und jede Anwendung definiert werden, welche Funktions- und Sicherheitsanforderungen an die Authentisierung gestellt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B24 **„Authentisierungsinformationen MÜSSEN kryptografisch sicher gespeichert werden.“**

B25 **„Authentisierungsinformationen DÜRFEN NICHT unverschlüsselt über unsichere Netze übertragen werden.“**

ORP.4.A13 Geeignete Auswahl von Authentisierungsmechanismen [IT-Betrieb] (S)

B26 **„Es SOLLTEN dem Schutzbedarf angemessene Identifikations- und Authentisierungsmechanismen verwendet werden.“**

B27 **„Authentisierungsdaten SOLLTEN durch das IT-System bzw. die IT-Anwendungen bei der Verarbeitung jederzeit gegen Ausspähung, Veränderung und Zerstörung geschützt werden.“**

B28 **„Das IT-System bzw. die IT-Anwendung SOLLTE nach jedem erfolglosen Authentisierungsversuch weitere Anmeldeversuche zunehmend verzögern (Time Delay)“**

B29 **„Die Gesamtdauer eines Anmeldeversuchs SOLLTE begrenzt werden können.“**

B30 **„Nach Überschreitung der vorgegebenen Anzahl erfolgloser Authentisierungsversuche SOLLTE das IT-System bzw. die IT-Anwendung die Benutzerkennung sperren.“**

ORP.4.A14 Kontrolle der Wirksamkeit der Benutzertrennung am IT-System bzw. an der Anwendung [IT-Betrieb] (S)

× **„In angemessenen Zeitabständen SOLLTE überprüft werden, ob die Benutzer von IT-Systemen bzw. Anwendungen sich regelmäßig nach Aufgabenerfüllung abmelden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt

werden muss.

B31 **„Ebenso SOLLTE kontrolliert werden, dass nicht mehrere Benutzer unter der gleichen Kennung arbeiten.“**

ORP.4.A21 Mehr-Faktor-Authentisierung [IT-Betrieb] (H)

B32 **„Es SOLLTE eine sichere Mehr-Faktor-Authentisierung, z. B. mit kryptografischen Zertifikaten, Chipkarten oder Token, zur Authentisierung verwendet werden.“**

CON.1 Kryptokonzept

CON.1.A1 Auswahl geeigneter kryptografischer Verfahren [Fachverantwortliche] (B)

B33 **„Es MÜSSEN geeignete kryptografische Verfahren ausgewählt werden. Dabei MUSS sichergestellt sein, dass etablierte Algorithmen verwendet werden, die von der Fachwelt intensiv untersucht wurden und von denen keine Sicherheitslücken bekannt sind.“**

B34 **„Ebenso MÜSSEN aktuell empfohlene Schlüssellängen verwendet werden.“**

CON.1.A3 Verschlüsselung der Kommunikationsverbindungen (S)

B35 **„Es SOLLTE geprüft werden, ob mit vertretbarem Aufwand eine Verschlüsselung der Kommunikationsverbindungen möglich und praktikabel ist. Ist dies der Fall, SOLLTEN Kommunikationsverbindungen geeignet verschlüsselt werden.“**

CON.1.A4 Geeignetes Schlüsselmanagement (S)

B36 **„Kryptografische Schlüssel SOLLTEN immer mit geeigneten Schlüsselgeneratoren und in einer sicheren Umgebung erzeugt werden.“**

B37 **„Ein Schlüssel SOLLTE möglichst nur einem Einsatzzweck dienen.“**

B38 **„Insbesondere SOLLTEN für die Verschlüsselung und Signaturbildung unterschiedliche Schlüssel benutzt werden.“**

B39 **„Der Austausch von kyptografischen Schlüsseln SOLLTE mit einem als sicher geltenden Verfahren durchgeführt werden.“**

B40 **„Wenn Schlüssel verwendet werden, SOLLTE die authentische Herkunft und die Integrität der Schlüsseldaten überprüft werden.“**

× **„Alle kryptografischen Schlüssel SOLLTEN hinreichend häufig gewechselt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Es SOLLTE eine festgelegte Vorgehensweise für den Fall geben, dass ein Schlüssel offengelegt wurde.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B41 **„Alle erzeugten kryptografischen Schlüssel SOLLTEN sicher aufbewahrt und verwaltet werden.“**

CON.1.A8 Erhebung der Einflussfaktoren für kryptografische Verfahren und Produkte (H)

B42 **„Bevor entschieden werden kann, welche kryptografischen Verfahren und Produkte bei erhöhtem Schutzbedarf eingesetzt werden, SOLLTEN unter anderem folgende Einflussfaktoren ermittelt werden:**

- **Sicherheitsaspekte,**
- **technische Aspekte,**
- **personelle und organisatorische Aspekte,**
- **wirtschaftliche Aspekte,**
- **Lebensdauer von kryptografischen Verfahren und der eingesetzten Schlüssellängen,**
- **Zulassung von kryptografischen Produkten sowie**
- **gesetzliche Rahmenbedingungen.“**

CON.1.A11 Sichere Konfiguration der Kryptomodule [IT-Betrieb] (H)

B43 **„Kryptomodule SOLLTEN sicher installiert und konfiguriert werden.“**

B44 **„Alle voreingestellten Schlüssel SOLLTEN geändert werden.“**

× **„Anschließend SOLLTE getestet werden, ob die Kryptomodule korrekt funktionieren und vom Benutzer auch bedient werden können.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt

werden muss.

× **„Weiterhin SOLLTEN die Anforderungen an die Einsatzumgebung festgelegt werden.“**
Entfällt, da eine anwendungsübergreifende Definition von Anforderungen an die Einsatzumgebung außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Wenn ein IT-System geändert wird, SOLLTE getestet werden, ob die eingesetzten kryptografischen Verfahren noch greifen.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Die Konfiguration der Kryptomodule SOLLTE dokumentiert und regelmäßig überprüft werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

CON.1.A13 Anforderungen an die Betriebssystem-Sicherheit beim Einsatz von Kryptomodulen (H)

B45 **„Das Zusammenwirken von Betriebssystem und Kryptomodulen SOLLTE gewährleisten, dass**

- **die installierten Kryptomodule nicht unbemerkt abgeschaltet oder umgangen werden können,**
- **die angewendeten oder gespeicherten Schlüssel nicht kompromittiert werden können,**
- **die zu schützenden Daten nur mit Wissen und unter Kontrolle des Benutzers auch unverschlüsselt auf Datenträgern abgespeichert werden bzw. das informationsverarbeitende System verlassen können sowie**
- **Manipulationsversuche am Kryptomodul erkannt werden.“**

CON.8 Software-Entwicklung

CON.8.A5 Sicheres Systemdesign (B)

B46 **„Grundsätzlich MÜSSEN alle Eingabedaten vor der Weiterverarbeitung geprüft und validiert werden.“**

× **„Bei Client-Server-Anwendungen MÜSSEN die Daten grundsätzlich**

auf dem Server validiert werden.“
Entfällt, da eine anwendungsübergreifende Validierung der Dateneingaben außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

B47 **„Die Standardeinstellungen der Software MÜSSEN derart voreingestellt sein, dass ein sicherer Betrieb der Software ermöglicht wird.“**

B48 **„Bei Fehlern oder Ausfällen von Komponenten des Systems DÜRFEN NICHT schützenswerte Informationen preisgegeben werden.“**

B49 **„Die Software MUSS mit möglichst geringen Privilegien ausgeführt werden können.“**

× **„Schützenswerte Daten MÜSSEN entsprechend der Vorgaben des Kryptokonzepts verschlüsselt übertragen und gespeichert werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B50 **„Zur Benutzer-Authentisierung und Authentifizierung MÜSSEN vertrauenswürdige Mechanismen verwendet werden, die den Sicherheitsanforderungen an die Anwendung entsprechen.“**

B51 **„Falls zur Authentifizierung Passwörter gespeichert werden, MÜSSEN diese mit einem sicheren Hashverfahren gespeichert werden.“**

B52 **„Sicherheitsrelevante Ereignisse MÜSSEN in der Art protokolliert werden, dass sie im Nachgang ausgewertet werden können.“**

× **„Informationen, die für den Produktivbetrieb nicht relevant sind (z. B. Kommentare mit Zugangsdaten für die Entwicklungsumgebung), SOLLTEN in ausgeliefertem Programmcode und ausgelieferten Konfigurationsdateien entfernt werden.“**
Entfällt, da eine Prüfung des Programmcodes außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Das Systemdesign MUSS dokumentiert werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Es MUSS überprüft werden, ob alle Sicherheitsanforderungen an das Systemdesign erfüllt wurden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

CON.8.A6 Verwendung von externen Bibliotheken aus vertrauenswürdigen Quellen (B)

B53 **„Wird im Rahmen des Entwicklungs- und Implementierungsprozesses auf externe Bibliotheken zurückgegriffen, MÜSSEN diese aus vertrauenswürdigen Quellen bezogen werden.“**

B54 **„Bevor externe Bibliotheken verwendet werden, MUSS deren Integrität sichergestellt werden.“**

CON.10 Entwicklung von Webanwendungen

CON.10.A9 Schutz vor SQL-Injection (B)

B55 **„Falls Daten an ein Datenbankmanagementsystem (DBMS) weitergeleitet werden, MÜSSEN die Entwickler Stored Procedures bzw. Prepared SQL Statements einsetzen.“**

× **„Falls Daten an ein DBMS weitergeleitet werden und weder Stored Procedures noch Prepared SQL Statements von der Einsatzumgebung unterstützt werden, MÜSSEN die SQL-Queries separat abgesichert werden.“**

Entfällt, da eine Absicherung von externen SQL-Queries außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

CON.10.A10 Restriktive Herausgabe sicherheitsrelevanter Informationen (B)

B56 **„Die Entwickler MÜSSEN sicherstellen, dass Webseiten, Rückantworten und Fehlermeldungen von Webanwendungen keine Informationen enthalten, die einem Angreifer Hinweise darauf geben, wie er Sicherheitsmechanismen umgehen kann.“**

OPS.1.1.3 Patch- und Änderungsmanagement

OPS.1.1.3.A3 Konfiguration von Autoupdate-Mechanismen (B)

× **„Innerhalb der Strategie zum Patch- und Änderungsmanagement MUSS definiert werden, wie mit integrierten Update-Mechanismen (Autoupdate) der eingesetzten Software umzugehen ist.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B57 **„Außerdem SOLLTEN neue Komponenten daraufhin überprüft werden, welche Update-Mechanismen sie haben. Insbesondere MUSS festgelegt werden, wie diese Mechanismen abgesichert und passend konfiguriert werden.“**

OPS.1.1.3.A10 Sicherstellung der Integrität und Authentizität von Softwarepaketen (S)

B58 **„Während des gesamten Patch- oder Änderungsprozesses SOLLTE die Authentizität und Integrität von Softwarepaketen sichergestellt werden. Dazu SOLLTE geprüft werden, ob für die eingesetzten Softwarepakete Prüfsummen oder digitale Signaturen verfügbar sind. Falls ja, SOLLTEN diese vor der Installation des Pakets überprüft werden.“**

× **„Ebenso SOLLTE darauf geachtet werden, dass die notwendigen Programme zur Überprüfung vorhanden sind.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B59 **„Software und Updates SOLLTEN grundsätzlich nur aus vertrauenswürdigen Quellen bezogen werden.“**

OPS.1.1.4 Schutz vor Schadprogrammen

OPS.1.1.4.A2 Nutzung systemspezifischer Schutzmechanismen (B)

B60 **„Es MUSS geprüft werden, welche Schutzmechanismen die verwendeten IT-Systeme sowie die darauf genutzten Betriebssysteme und Anwendungen bieten. Diese Mechanismen MÜSSEN genutzt werden, sofern es keinen mindestens gleichwertigen Ersatz gibt oder gute Gründe dagegen sprechen.“**

× **„Werden sie nicht genutzt, MUSS dies begründet und dokumentiert werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

OPS.1.1.5 Protokollierung

OPS.1.1.5.A3 Konfiguration der Protokollierung auf System- und Netzebene (B)

B61 **„Alle sicherheitsrelevanten Ereignisse von IT-Systemen und Anwendungen MÜSSEN protokolliert werden.“**

× **„Sofern die in der Protokollierungsrichtlinie als relevant definierten IT-Systeme und Anwendungen über eine Protokollierungsfunktion verfügen, MUSS diese benutzt werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B62 **„Wenn die Protokollierung eingerichtet wird, MÜSSEN dabei die Herstellervorgaben für die jeweiligen IT-Systeme oder Anwendungen beachtet werden.“**

× **„In angemessenen Intervallen MUSS stichpunktartig überprüft werden, ob die Protokollierung noch korrekt funktioniert. Die Prüfintervalle MÜSSEN in der Protokollierungsrichtlinie definiert werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Falls betriebs- und sicherheitsrelevante Ereignisse nicht auf einem IT-System protokolliert werden können, MÜSSEN zusätzliche IT-Systeme zur Protokollierung (z. B. von Ereignissen auf Netzebene) integriert werden.“**

Entfällt, da die Bereitstellung von zusätzlichen IT-Systemen zur Protokollierung außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

OPS.1.1.5.A4 Zeitsynchronisation der IT-Systeme (B)

B63 **„Die Systemzeit aller protokollierenden IT-Systeme und Anwendungen MUSS immer synchron sein.“**

B64 **„Es MUSS sichergestellt sein, dass das Datums- und Zeitformat der Protokolldateien einheitlich ist.“**

OPS.1.1.5.A5 Einhaltung rechtlicher Rahmenbedingungen (B)

× **„Bei der Protokollierung MÜSSEN die Bestimmungen aus den aktuellen Gesetzen zum Bundes- sowie Landesdatenschutz eingehalten werden.“**
Entfällt, da die Berücksichtigung von Datenschutzfragen außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Darüber hinaus MÜSSEN eventuelle Persönlichkeitsrechte bzw. Mitbestimmungsrechte der Mitarbeitervertretungen gewahrt werden.“**
Entfällt, da die Berücksichtigung von Persönlichkeitsrechten außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Ebenso MUSS sichergestellt sein, dass alle weiteren relevanten gesetzlichen Bestimmungen beachtet werden.“**
Entfällt, da die Berücksichtigung von gesetzlichen Bestimmungen außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Protokollierungsdaten MÜSSEN nach einem festgelegten Prozess gelöscht werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B65 **„Es MUSS technisch unterbunden werden, dass Protokollierungsdaten unkontrolliert gelöscht oder verändert werden.“**

OPS.1.1.5.A10 Zugriffsschutz für Protokollierungsdaten (S)

B66 **„Es SOLLTE sichergestellt sein, dass die ausführenden Administratoren selbst keine Berechtigung haben, die aufgezeichneten Protokollierungsdaten zu verändern oder zu löschen.“**

OPS.1.1.5.A11 Steigerung des Protokollierungsumfangs (H)

B67 **„Bei erhöhtem Schutzbedarf von Anwendungen oder IT-Systemen SOLLTEN grundsätzlich mehr Ereignisse protokolliert werden, sodass sicherheitsrelevante Vorfälle möglichst lückenlos nachvollziehbar sind.“**

× **„Um die Protokollierungsdaten in Echtzeit auswerten zu können, SOLLTEN sie in verkürzten Zeitabständen von den protokollierenden IT-Systemen und Anwendungen zentral gespeichert werden.“**

Entfällt, da der Export/die Synchronisierung von Protokollierungsdaten außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Die Protokollierung SOLLTE eine Auswertung über den gesamten Informationsverbund ermöglichen.“**
Entfällt, da eine anwendungsübergreifende Protokollierung außerhalb des betrachteten DBMS liegt und daher nicht Teil dieser Master-Thesis ist.

× **„Anwendungen und IT-Systeme, mit denen eine zentrale Protokollierung nicht möglich ist, SOLLTEN bei einem erhöhten Schutzbedarf NICHT eingesetzt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.4.3 Relationale Datenbanksysteme

APP.4.3.A9 Datensicherung eines Datenbanksystems (B)

B68 **„Es MÜSSEN regelmäßig Systemsicherungen des DBMS und der Daten durchgeführt werden. Auch bevor eine Datenbank neu erzeugt wird, MUSS das Datenbanksystem gesichert werden. Hierfür SOLLTEN die dafür zulässigen Dienstprogramme benutzt werden.“**

B69 **„Alle Transaktionen SOLLTEN so gesichert werden, dass sie jederzeit wiederherstellbar sind.“**

× **„Wenn die Datensicherung die verfügbaren Kapazitäten übersteigt, SOLLTE ein erweitertes Konzept erstellt werden, um die Datenbank zu sichern, z. B. eine inkrementelle Sicherung.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Abhängig vom Schutzbedarf der Daten SOLLTEN die Wiederherstellungsparameter vorgegeben werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.4.3.A11 Ausreichende Dimensionierung der Hardware [Fachverantwortliche] (S)

B70 **„Datenbankmanagementsysteme SOLLTEN auf ausreichend dimensionierter Hardware installiert werden. Die Hardware SOLLTE über genügend Reserven verfügen, um auch eventuell steigenden Anforderungen gerecht zu werden.“**

× **„Zeichnen sich trotzdem während des Betriebs Ressourcenengpässe ab, SOLLTEN diese frühzeitig behoben werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Wenn die Hardware dimensioniert wird, SOLLTE das erwartete Wachstum für den geplanten Einsatzzeitraum berücksichtigt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.4.3.A13 Restriktive Handhabung von Datenbank-Links (S)

B71 **„Es SOLLTE sichergestellt sein, dass nur Verantwortliche dazu berechtigt sind, Datenbank-Links (DBLinks) anzulegen.“**

B72 **„Werden solche Links angelegt, MÜSSEN so genannte Private DB-Links vor Public DB-Links bevorzugt angelegt werden.“**

× **„Alle von den Verantwortlichen angelegten DB-Links SOLLTEN dokumentiert und regelmäßig überprüft werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Zudem SOLLTEN DB-Links mitberücksichtigt werden, wenn das Datenbanksystem gesichert wird.“**
Entfällt, da die Berücksichtigung von Datensicherungsverfahren als externe Maßnahme nicht Teil dieser Master-Thesis ist.

APP.4.3.A16 Verschlüsselung der Datenbankanbindung (S)

B73 **„Das Datenbankmanagementsystem SOLLTE so konfiguriert werden, dass Datenbankverbindungen immer verschlüsselt werden.“**

× **„Die dazu eingesetzten kryptografischen Verfahren und Protokolle SOLLTEN den internen Vorgaben der Institution entsprechen.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.4.3.A18 Überwachung des Datenbankmanagementsystems (S)

B74 **„Die für den sicheren Betrieb kritischen Parameter, Ereignisse und Betriebszustände des Datenbankmanagementsystems SOLLTEN definiert werden. Diese SOLLTEN mithilfe eines Monitoring-Systems überwacht werden.“**

B75 **„Für alle kritischen Parameter, Ereignisse und Betriebszustände SOLLTEN Schwellwerte festgelegt werden. Wenn diese Werte überschritten werden, MUSS geeignet reagiert werden. Hierbei SOLLTEN die zuständigen Mitarbeiter alarmiert werden.“**

× **„Anwendungsspezifische Parameter, Ereignisse, Betriebszustände und deren Schwellwerte SOLLTEN mit den Verantwortlichen für die Fachanwendungen abgestimmt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.4.3.A19 Schutz vor schädlichen Datenbank-Skripten [Entwickler] (S)

× **„Werden Datenbank-Skripte entwickelt, SOLLTEN dafür verpflichtende Qualitätskriterien definiert werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B76 **„Datenbank-Skripte SOLLTEN ausführlichen Funktionstests auf gesonderten Testsystemen unterzogen werden, bevor sie produktiv eingesetzt werden.“**

× **„Die Ergebnisse SOLLTEN dokumentiert werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.4.3.A24 Datenverschlüsselung in der Datenbank (H)

B77 **„Die Daten in den Datenbanken SOLLTEN verschlüsselt werden. Dabei SOLLTEN vorher unter anderem folgende Faktoren betrachtet werden:**

- **Einfluss auf die Performance,**
- **Schlüsselverwaltungsprozesse und -verfahren, einschließlich separater Schlüsselaufbewahrung und -sicherung,**
- **Einfluss auf Backup-Recovery-Konzepte,**
- **funktionale Auswirkungen auf die Datenbank, beispielsweise Sortiermöglichkeiten.“**

APP.6 Allgemeine Software

APP.6.A3 Sichere Beschaffung von Software [Beschaffungsstelle] (B)

× **„Wenn Software beschafft wird, MUSS auf Basis des Anforderungskatalog eine geeignete Software ausgewählt werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

B78 **„Die ausgewählte Software MUSS aus vertrauenswürdigen Quellen beschafft werden.“**

B79 **„Die vertrauenswürdige Quelle SOLLTE eine Möglichkeit bereitstellen, die Software auf Integrität zu überprüfen.“**

× **„Darüber hinaus SOLLTE die Software mit einem geeigneten Wartungsvertrag oder einer vergleichbaren Zusage des Herstellers oder Software-Anbieters beschafft werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Diese Verträge oder Zusagen SOLLTEN insbesondere garantieren, dass auftretende Sicherheitslücken und Schwachstellen der Software während des gesamten Nutzungszeitraums zeitnah behoben werden.“**
Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.6.A4 Regelung für die Installation und Konfiguration von Software [Fachverantwortliche] (B)

B80 **„Die Installation und Konfiguration der Software MUSS durch den IT-Betrieb so geregelt werden, dass**

- **die Software nur mit dem geringsten notwendigen Funktions-**

umfang installiert und ausgeführt wird,

- **die Software mit den geringsten möglichen Berechtigungen ausgeführt wird,**
- **die datensparsamsten Einstellungen (in Bezug auf die Verarbeitung von personenbezogenen Daten) konfiguriert werden sowie**
- **alle relevanten Sicherheitsupdates und -patches installiert sind, bevor die Software produktiv eingesetzt wird.“**

„Hierbei MÜSSEN auch abhängige Komponenten (u. a. Laufzeitumgebungen, Bibliotheken, Schnittstellen sowie weitere Programme) mitbetrachtet werden.“

× **„Der IT-Betrieb MUSS in Abstimmung mit dem Fachverantwortlichen festlegen, wer die Software wie installieren darf. Idealerweise SOLLTE Software immer zentral durch den IT-Betrieb installiert werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Ist es erforderlich, dass die Software (teilweise) manuell installiert wird, dann MUSS der IT-Betrieb eine Installationsanweisung erstellen, in der klar geregelt wird, welche Zwischenschritte zur Installation durchzuführen und welche Konfigurationen vorzunehmen sind.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Darüber hinaus MUSS der IT-Betrieb regeln, wie die Integrität der Installationsdateien überprüft wird. Falls zu einem Installationspaket digitale Signaturen oder Prüfsummen verfügbar sind, MÜSSEN mit diesen die Integrität überprüft werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Sofern erforderlich, SOLLTE der IT-Betrieb eine sichere Standardkonfiguration der Software festlegen, mit der die Software konfiguriert wird. Die Standardkonfiguration SOLLTE dokumentiert werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

APP.6.A11 Verwendung von Plug-ins und Erweiterungen (S)

B81 **„Es SOLLTEN nur unbedingt notwendige Plug-ins und Erweiterungen installiert werden.“**

B82 **„Werden Erweiterungen eingesetzt, SOLLTE die Software die Möglichkeit bieten, Erweiterungen zu konfigurieren und abzuschalten.“**

G Identifikation aller Einzelanforderungen der CIS-Benchmarks

Nachfolgend werden alle anwendbaren Maßnahmen der genutzten CIS-Bechmarks [79, 80, 81] mit einer fortlaufenden Nummer mit dem Präfix „C“ versehen, um eine Gruppierung und Referenzierung zu ermöglichen. Alle Maßnahmen, die nicht berücksichtigt werden, enthalten eine entsprechende Begründung.

CIS Apache Cassandra 3.11 Benchmark

Installation und Updates

C1 **„Stellen Sie sicher, dass ein eigener Benutzer und eine eigene Gruppe für Cassandra existieren.“**

× **„Stellen Sie sicher, dass die neueste Version von Java installiert ist.“**
Entfällt, da die Überprüfung des Patch-Status nicht über eine technische Konfigurationsanpassung abgebildet werden kann, sondern durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Stellen Sie sicher, dass die aktuellste Version von Python installiert ist.“**
Entfällt, da die Überprüfung des Patch-Status nicht über eine technische Konfigurationsanpassung abgebildet werden kann, sondern durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

× **„Stellen Sie sicher, dass die aktuellste Version von Cassandra installiert ist.“**
Entfällt, da die Überprüfung des Patch-Status nicht über eine technische Konfigurationsanpassung abgebildet werden kann, sondern durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

C2 **„Stellen Sie sicher, dass der Cassandra-Dienst nicht als root-Benutzer ausgeführt wird.“**

C3 **„Stellen Sie sicher, dass die Uhrzeit auf allen Nodes synchronisiert ist.“**

Authentifizierung und Autorisierung

C4 **„Stellen Sie sicher, dass die Authentifizierung für Cassandra-Datenbanken aktiviert ist.“**

C5 **„Stellen Sie sicher, dass die Autorisierung für Cassandra-Datenbanken aktiviert ist.“**

Zugangskontrolle/Passwortrichtlinien

C6 **„Stellen Sie sicher, dass die Cassandra- und Superuser-Rollen getrennt sind.“**

C7 **„Stellen Sie sicher, dass das Standardpasswort für die Cassandra-Rolle geändert wurde.“**

C8 **„Stellen Sie sicher, dass es keine unnötigen Rollen oder übermäßigen Privilegien vorhanden sind.“**

C9 **„Stellen Sie sicher, dass Cassandra mit einem nicht privilegierten, dedizierten Dienstkonto ausgeführt wird.“**

C10 **„Stellen Sie sicher, dass Cassandra nur Netzwerkverbindungen von autorisierten Schnittstellen empfängt.“**

C11 **„Überprüfen Sie die benutzerdefinierten Rollen.“**

C12 **„Überprüfen Sie Superuser/Admin-Rollen.“**

Auditierung und Protokollierung

C13 **„Stellen Sie sicher, dass die Protokollierung aktiviert ist.“**

C14 **„Stellen Sie sicher, dass die Audit-Funktion aktiviert ist.“**

Verschlüsselung

C15 **„Stellen Sie sicher, dass die Inter-Node-Verschlüsselung aktiviert ist.“**

C16 **„Stellen Sie sicher, dass die Client-Verschlüsselung aktiviert ist.“**

CIS MongoDB 5 Benchmark

Installation und Patching

× **„Stellen Sie sicher, dass die entsprechende MongoDB-Softwareversion bzw. die entsprechenden Patches installiert sind.“**
Entfällt, da die Überprüfung des Patch-Status nicht über eine technische Konfigurationsanpassung abgebildet werden kann, sondern durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

Authentifizierung

C17 **„Stellen Sie sicher, dass die Authentifizierung konfiguriert ist.“**
C18 **„Stellen Sie sicher, dass MongoDB die Authentifizierung nicht über den Localhost umgeht.“**
C19 **„Stellen Sie sicher, dass die Authentifizierung im Sharded-Cluster aktiviert ist.“**

Autorisierung

C20 **„Stellen Sie sicher, dass für Datenbankkonten die geringsten Rechte gelten.“**
C21 **„Stellen Sie sicher, dass die rollenbasierte Zugriffskontrolle aktiviert und entsprechend konfiguriert ist.“**
C22 **„Stellen Sie sicher, dass MongoDB mit einem nicht privilegierten, dedizierten Dienstkonto ausgeführt wird.“**
C23 **„Stellen Sie sicher, dass jede Rolle für jede MongoDB-Datenbank erforderlich ist und nur die notwendigen Privilegien gewährt.“**
C24 **„Überprüfen Sie die Superuser/Admin-Rollen.“**

Datenverschlüsselung

C25 **„Stellen Sie sicher, dass veraltete TLS-Protokolle deaktiviert sind.“**
C26 **„Stellen Sie sicher, dass schwache Protokolle deaktiviert sind.“**
C27 **„Stellen Sie sicher, dass die Daten während der Übertragung mit TLS oder SSL (Transportverschlüsselung) verschlüsselt werden.“**
C28 **„Stellen Sie sicher, dass der Federal Information Processing Standard (FIPS) aktiviert ist.“**

C29 **„Stellen Sie die Verschlüsselung von Daten im ruhendem Zustand sicher.“**

Audit-Protokollierung

C30 **„Stellen Sie sicher, dass die Systemaktivität überwacht wird.“**

C31 **„Stellen Sie sicher, dass Audit-Filter richtig konfiguriert sind.“**

C32 **„Stellen Sie sicher, dass die Protokollierung so viele Informationen wie möglich erfasst.“**

C33 **„Stellen Sie sicher, dass neue Einträge an das Ende der Protokolldatei angehängt werden.“**

Härtung des Betriebssystems

C34 **„Stellen Sie sicher, dass MongoDB nicht den Standard-Port verwendet.“**

C35 **„Stellen Sie sicher, dass die Betriebssystem-Ressourcenlimits für MongoDB festgelegt sind.“**

C36 **„Stellen Sie sicher, dass das serverseitige Scripting deaktiviert ist, sofern es nicht benötigt wird.“**

Dateiberechtigungen

C37 **„Stellen Sie sicher, dass die geeigneten Berechtigungen für Schlüsseldateien gesetzt sind.“**

C38 **„Stellen Sie sicher, dass die geeigneten Dateiberechtigungen für die Datenbanken gesetzt sind.“**

CIS PostgreSQL 14 Benchmark

Installation und Patches

C39 **„Stellen Sie sicher, dass die Softwarepakete von autorisierten Repositories bezogen werden.“**

C40 **„Stellen Sie sicher, dass die systemd-Dienstdateien aktiviert sind.“**

C41 **„Stellen Sie sicher, dass das Daten-Cluster erfolgreich initialisiert wurde.“**

Verzeichnis- und Dateiberechtigungen

C42 **„Stellen Sie sicher, dass die Dateiberechtigungen ordnungsgemäß sind.“**

Protokollierung, Monitoring und Auditing

C43 **„Stellen Sie sicher, dass die Methoden für die Protokollierung richtig eingestellt sind.“**

C44 **„Stellen Sie sicher, dass der Logging-Collector aktiviert ist.“**

C45 **„Stellen Sie sicher, dass das Zielverzeichnis für die Protokolldateien richtig eingestellt ist.“**

C46 **„Stellen Sie sicher, dass das Dateinamensmuster für die Protokolldateien korrekt eingestellt ist.“**

C47 **„Stellen Sie sicher, dass die Berechtigungen für die Protokolldateien richtig eingestellt sind.“**

C48 **„Stellen Sie sicher, dass 'log_truncate_on_rotation' aktiviert ist.“**

C49 **„Stellen Sie sicher, dass die maximale Lebensdauer der Protokolldatei korrekt eingestellt ist.“**

C50 **„Stellen Sie sicher, dass die maximale Größe der Protokolldatei richtig eingestellt ist.“**

C51 **„Stellen Sie sicher, dass die richtige Syslog-Funktion ausgewählt ist.“**

C52 **„Stellen Sie sicher, dass Syslog-Meldungen nicht unterdrückt werden.“**

C53 **„Stellen Sie sicher, dass Syslog-Meldungen nicht aufgrund ihrer Länge verloren gehen.“**

C54 **„Stellen Sie sicher, dass der Programmname für PostgreSQL-Syslog-Meldungen korrekt ist.“**

C55 **„Stellen Sie sicher, dass die gewünschten Meldungen in das Serverprotokoll geschrieben werden.“**

C56 **„Stellen Sie sicher, dass SQL-Anweisungen, die Fehler erzeugen, aufgezeichnet werden.“**

C57 **„Stellen Sie sicher, dass 'debug_print_parse' deaktiviert ist.“**

C58 **„Stellen Sie sicher, dass 'debug_print_rewritten' deaktiviert ist.“**

C59 **„Stellen Sie sicher, dass 'debug_print_plan' deaktiviert ist.“**

C60 **„Stellen Sie sicher, dass 'debug_pretty_print' aktiviert ist.“**

C61 **„Stellen Sie sicher, dass 'log_connections' aktiviert ist.“**

C62 **„Stellen Sie sicher, dass 'log_disconnections' aktiviert ist.“**

C63 **„Stellen Sie sicher, dass 'log_error_verbosity' korrekt eingestellt ist.“**

C64 **„Stellen Sie sicher, dass 'log_hostname' korrekt eingestellt ist.“**

C65 **„Stellen Sie sicher, dass 'log_line_prefix' richtig eingestellt ist.“**

C66 **„Stellen Sie sicher, dass 'log_statement' richtig eingestellt ist.“**

C67 **„Stellen Sie sicher, dass 'log_timezone' richtig eingestellt ist.“**

C68 **„Stellen Sie sicher, dass die PostgreSQL-Audit-Extension (pgAudit) aktiviert ist.“**

Benutzerzugriff und Autorisierung

C69 **„Stellen Sie sicher, dass sudo richtig konfiguriert ist.“**

C70 **„Stellen Sie sicher, dass überschüssige administrative Privilegien entzogen werden.“**

C71 **„Stellen Sie sicher, dass überschüssige Funktionsberechtigungen entzogen werden.“**

C72 **„Stellen Sie sicher, dass überschüssige DML-Berechtigungen entzogen werden.“**

C73 **„Stellen Sie sicher, dass RLS richtig konfiguriert ist.“**

C74 **„Stellen Sie sicher, dass die 'set_user'-Erweiterung installiert ist.“**

C75 **„Verwenden Sie die vordefinierten Benutzerrollen.“**

Verbindung und Authentifizierung

C76 **„Stellen Sie sicher, dass die Anmeldung über einen lokalen UNIX-Domain-Socket ordnungsgemäß konfiguriert ist.“**

C77 **„Stellen Sie sicher, dass die Anmeldung über den Host-TCP/IP-Socket korrekt konfiguriert ist.“**

PostgreSQL-Einstellungen

× **„Stellen Sie sicher, dass die Backend-Laufzeitparameter ordnungsgemäß konfiguriert sind.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass die Postmaster-Laufzeitparameter ordnungsgemäß konfiguriert sind.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass die SIGHUP-Laufzeitparameter ordnungsgemäß konfiguriert sind.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass die Superuser-Laufzeitparameter ordnungsgemäß konfiguriert sind.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass die User-Laufzeitparameter ordnungsgemäß konfiguriert sind.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

C78 **„Stellen Sie sicher, dass FIPS 140-2 OpenSSL-Kryptographie verwendet wird.“**

C79 **„Stellen Sie sicher, dass TLS aktiviert und korrekt konfiguriert ist.“**

C80 **„Stellen Sie sicher, dass die 'pgcrypto'-Erweiterung installiert und ordnungsgemäß konfiguriert ist.“**

Replikation

C81 **„Stellen Sie sicher, dass ein alleiniger Replikationsbenutzer erstellt und für die Streaming-Replikation verwendet wird.“**

C82 **„Stellen Sie sicher, dass die Protokollierung von Replikationsbefehlen konfiguriert ist.“**

C83 **„Stellen Sie sicher, dass Basis-Backups konfiguriert sind und funktionieren.“**

C84 **„Stellen Sie sicher, dass die WAL-Archivierung konfiguriert und funktionsfähig ist.“**

C85 **„Stellen Sie sicher, dass die Streaming-Replikationsparameter korrekt konfiguriert sind.“**

Besondere Konfigurationshinweise

× **„Stellen Sie sicher, dass sich die PostgreSQL-Konfigurationsdateien außerhalb des Daten-Clusters befinden.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass sich die PostgreSQL-Unterverzeichnisse außerhalb des Daten-Clusters befinden.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass das Sicherungs- und Wiederherstellungswerkzeug 'pgBackRest' installiert und konfiguriert ist.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

× **„Stellen Sie sicher, dass die sonstigen Konfigurationseinstellungen korrekt sind.“**
Entfällt, da es sich um Konfigurationsanpassungen handelt, die nur speziell für PostgreSQL umgesetzt werden können.

H Identifikation aller Einzelanforderungen der STIGs-Standards

Nachfolgend werden alle anwendbaren Maßnahmen der genutzten STIGs-Leitfäden [82, 83] mit einer fortlaufenden Nummer mit dem Präfix „S“ versehen, um eine Gruppierung und Referenzierung zu ermöglichen. Alle Maßnahmen, die nicht berücksichtigt werden, enthalten eine entsprechende Begründung.

MongoDB Enterprise Advanced 4.x STIG Version 1, Release 1

S1 **„MongoDB muss in allen DBMS-/Datenbank-Komponenten die Erstellung von Audit-Protokollen für DoD-definierte auditierbare Ereignisse ermöglichen.“**

S2 **„Die von MongoDB erzeugten Audit-Informationen müssen vor unberechtigtem Zugriff geschützt werden.“**

S3 **„MongoDB muss seine Audit-Funktionen vor unautorisiertem Zugriff schützen.“**

S4 **„Nicht verwendete Datenbankkomponenten, die in MongoDB integriert sind und nicht deinstalliert werden können, müssen deaktiviert werden.“**

S5 **„Wenn Passwörter zur Authentifizierung verwendet werden, darf MongoDB nur verschlüsselte Darstellungen von Passwörtern übertragen.“**

S6 **„MongoDB muss nicht-organisatorische Benutzer (oder Prozesse, die im Namen von nicht-organisatorischen Benutzern handeln) eindeutig identifizieren und authentifizieren.“**

S7 **„MongoDB muss in einen sicheren Zustand übergehen, wenn die Systeminitialisierung fehlschlägt, das Herunterfahren fehlschlägt oder Abbrüche fehlschlagen.“**

S8 **„MongoDB muss die unbefugte und unbeabsichtigte Übertragung von Informationen über gemeinsam genutzte Systemressourcen verhindern.“**

S9 **„MongoDB und die zugehörigen Anwendungen müssen die Verwendung von dynamischer Codeausführung für Situationen vorbehalten, die dies erfordern.“**

× **„MongoDB muss unternehmensdefinierte Typen von Sicherheitskennzeichen mit unternehmensdefinierten Sicherheitskennzeichenwerten mit Informationen bei der Speicherung und Übertragung verknüpfen.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S10 **„MongoDB muss diskretionäre Zugriffskontrollrichtlinien, wie vom Dateneigentümer definiert, für definierte Subjekte und Objekte durchsetzen.“**

S11 **„MongoDB muss NIST FIPS 140-2-validierte kryptografische Module für kryptografische Operationen verwenden.“**

S12 **„MongoDB muss kryptografische Mechanismen implementieren, um eine unbefugte Änderung der vom Unternehmen definierten ruhenden Informationen (die mindestens PII und klassifizierte Informationen umfassen) auf den vom Unternehmen definierten Informationssystemkomponenten zu verhindern.“**

S13 **„MongoDB muss die Gesamtzahl der gleichzeitigen Verbindungen zur Datenbank begrenzen.“**

× **„MongoDB muss in einen Authentifizierungs-/Zugriffsmechanismus auf Unternehmensebene integriert werden, der die Verwaltung und Automatisierung von Konten für alle Benutzer, Gruppen, Rollen und andere Auftraggeber ermöglicht.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S14 **„MongoDB muss genehmigte Berechtigungen für den logischen Zugriff auf Informationen und Systemressourcen in Übereinstimmung mit den geltenden Zugriffskontrollrichtlinien durchsetzen.“**

S15 **„MongoDB muss die Berechtigungen zum Ändern von Softwaremodulen einschränken, einschließlich gespeicherter Prozeduren, Funktionen und Triggern sowie Links zu Software außerhalb von MongoDB.“**

S16 **„Das MongoDB-Software-Installationskonto muss auf autorisierte Benutzer beschränkt sein.“**

S17 **„Die Datenbanksoftware, einschließlich der DBMS-Konfigurationsdateien,**

muss in speziellen Verzeichnissen oder DASD-Pools gespeichert werden, die vom Host-Betriebssystem und anderen Anwendungen getrennt sind.“

S18 **„Datenbankobjekte (einschließlich, aber nicht beschränkt auf Tabellen, Indizes, Speicher, gespeicherte Prozeduren, Funktionen, Trigger, Links zu Software außerhalb von MongoDB usw.) müssen Eigentum von Datenbank-/DBMS-Principals sein, die zum Eigentum berechtigt sind.“**

S19 **„Die Rolle(n)/Gruppe(n), die zur Änderung der Datenbankstruktur (einschließlich, aber nicht unbedingt beschränkt auf Tabellen, Indizes, Speicher usw.) und der Logikmodule (gespeicherte Prozeduren, Funktionen, Trigger, Links zu Software außerhalb von MongoDB usw.) verwendet werden, müssen auf autorisierte Benutzer beschränkt sein.“**

S20 **„MongoDB muss organisatorische Benutzer (oder Prozesse, die im Namen von organisatorischen Benutzern handeln) eindeutig identifizieren und authentifizieren.“**

S21 **„Wenn Passwörter für die Authentifizierung verwendet werden, muss MongoDB LDAP oder Kerberos für die Authentifizierung implementieren, um die DoD-Standards für Passwortkomplexität und -lebensdauer durchzusetzen.“**

S22 **„Wenn Passwörter für die Authentifizierung verwendet werden, darf MongoDB nur gehashte, gesalzene Darstellungen von Passwörtern speichern.“**

S23 **„MongoDB muss einen autorisierten Zugriff auf alle von MongoDB gespeicherten/verwendeten privaten PKI-Schlüssel erzwingen.“**

× **„MongoDB muss die PKI-authentifizierte Identität auf ein zugehöriges Benutzerkonto abbilden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S24 **„MongoDB muss die Rückmeldung von Authentifizierungsinformationen während des Authentifizierungsprozesses verbergen, um die Informationen vor einer möglichen Ausnutzung/Verwendung durch nicht autorisierte Personen zu schützen.“**

S25 **„MongoDB muss die Authentizität von Kommunikationssitzungen aufrechterhalten, indem es vor Man-in-the-Middle (MITM)-Angriffen schützt, die Sitzungs-ID-Werte erraten.“**

S26 **„MongoDB muss die Vertraulichkeit und Integrität aller Informationen im laufenden Betrieb schützen.“**

× **„Der Datenbankinhalt muss durch die Durchsetzung einer Datenübertragungsrichtlinie vor unbefugter und unbeabsichtigter Informationsübertragung geschützt werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S27 **„MongoDB muss die Gültigkeit aller Dateneingaben prüfen, mit Ausnahme derjenigen, die von der Organisation speziell festgelegt wurden.“**

S28 **„MongoDB muss nicht-privilegierten Benutzern Fehlermeldungen zur Verfügung stellen, die Informationen für Korrekturmaßnahmen liefern, ohne Informationen preiszugeben, die von Angreifern ausgenutzt werden könnten.“**

S29 **„MongoDB darf detaillierte Fehlermeldungen nur dem ISSO, ISSM, SA und DBA offenbaren.“**

S30 **„MongoDB muss eine Benutzersitzung automatisch beenden, wenn organisationsdefinierte Bedingungen oder Trigger-Ereignisse ein Trennen der Sitzung erfordern.“**

S31 **„MongoDB muss eine zentrale Verwaltung des Inhalts der von allen MongoDB-Komponenten erzeugten Audit-Datensätze nutzen.“**

S32 **„MongoDB muss die Speicherkapazität für Audit-Datensätze in Übereinstimmung mit den Speicheranforderungen für Audit-Datensätze am Standort zuweisen.“**

S33 **„MongoDB muss den zuständigen Support-Mitarbeitern eine Warnung zukommen lassen, wenn das zugewiesene Speichervolumen für Prüfprotokolle 75 Prozent der maximalen Speicherkapazität für Prüfprotokolle erreicht.“**

S34 **„MongoDB muss die Installation von Logikmodulen (gespeicherte Prozeduren, Funktionen, Trigger, Views usw.) durch Benutzer ohne expliziten privilegierten Status verbieten.“**

S35 **„MongoDB muss Zugriffsbeschränkungen im Zusammenhang mit Änderungen an der Konfiguration von MongoDB oder der Datenbank(en) durchsetzen.“**

S36 **„MongoDB muss von den Benutzern eine erneute Authentifizierung verlangen, wenn vom Unternehmen definierte Umstände oder Situationen eine erneute Authentifizierung erfordern.“**

S37 **„MongoDB muss die Verwendung von zwischengespeicherten Authentifikatoren nach einer von der Organisation festgelegten Zeitspanne verbieten.“**

S38 **„MongoDB darf nur Endteilnehmerzertifikate akzeptieren, die von DoD PKI oder DoD-zugelassenen PKI-Zertifizierungsstellen (CAs) für die Einrichtung aller verschlüsselten Sitzungen ausgestellt wurden.“**

S39 **„MongoDB muss die Vertraulichkeit und Integrität von Informationen während der Vorbereitung der Übertragung wahren.“**

S40 **„MongoDB muss die Vertraulichkeit und Integrität von Informationen während des Empfangs sicherstellen.“**

S41 **„Wenn ungültige Eingaben empfangen werden, muss sich MongoDB in einer vorhersehbaren und dokumentierten Weise verhalten, die den Unternehmens- und Systemzielen entspricht.“**

S42 **„Wenn Updates auf die MongoDB-Software angewendet werden, müssen alle Softwarekomponenten, die ersetzt oder überflüssig gemacht wurden, entfernt werden.“**

S43 **„Sicherheitsrelevante Software-Updates für MongoDB müssen innerhalb des von einer maßgeblichen Quelle (z. B. IAVM, CTOs, DTMs und STIGs) vorgegebenen Zeitraums installiert werden.“**

S44 **„MongoDB-Produkte müssen eine vom Hersteller unterstützte Version aufweisen.“**

S45 **„MongoDB muss gemäß den Sicherheitskonfigurationseinstellungen konfiguriert werden, die auf den DoD-Sicherheitskonfigurations- und -Implementierungsrichtlinien basieren, einschließlich STIGs, NSA-Konfigurationsleitfäden, CTOs, DTMs und IAVMs.“**

Redis Enterprise 6.x STIG Version 1, Release 1

S46 **„Redis Enterprise DBMS muss die Anzahl der gleichzeitigen Sitzungen auf eine vom Unternehmen festgelegte Anzahl pro Benutzer für alle Konten und/oder Kontotypen begrenzen.“**

× **„Redis Enterprise DBMS muss in einen Authentifizierungs-/Zugriffsmechanismus auf Unternehmensebene integriert werden, der eine Kontoverwaltung und -automatisierung für alle Benutzer, Gruppen, Rollen und andere Prinzipale bietet.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S47 **„Redis Enterprise DBMS muss genehmigte Berechtigungen für den logischen Zugriff auf Informationen und Systemressourcen in Übereinstimmung mit den geltenden Zugriffskontrollrichtlinien durchsetzen.“**

S48 **„Redis Enterprise DBMS muss diskretionäre Zugriffskontrollrichtlinien, wie vom Dateneigentümer definiert, für definierte Subjekte und Objekte durchsetzen.“**

S49 **„Redis Enterprise DBMS muss Zugriffskontrolllisten, wie vom Dateneigentümer definiert, für bestimmte Subjekte und Objekte durchsetzen.“**

S50 **„Redis Enterprise DBMS muss verhindern, dass nicht privilegierte Benutzer privilegierte Funktionen ausführen, einschließlich der Deaktivierung, Umgehung oder Änderung von implementierten Sicherheitsvorkehrungen/Gegenmaßnahmen.“**

S51 **„Die Ausführung von Softwaremodulen (einschließlich gespeicherter Prozeduren, Funktionen und Triggern) mit erhöhten Rechten muss auf die notwendigen Fälle beschränkt werden.“**

S52 **„Redis Enterprise DBMS muss die Möglichkeit bieten, Audit-Protokolle für DoD-definierte auditierbare Ereignisse innerhalb aller DBMS/Datenbankkomponenten zu erstellen.“**

S53 **„Redis Enterprise DBMS muss es nur dem ISSM (oder vom ISSM ernannten Personen oder Rollen) erlauben, auszuwählen, welche auditfähigen Ereignisse auditiert werden sollen.“**

S54 **„Redis Enterprise DBMS muss Audit-Protokolle für alle direkten Zugriffe auf die Datenbank(en) erstellen.“**

S55 **„Redis Enterprise DBMS muss eine zentrale Verwaltung des Inhalts**

der von allen Komponenten von Redis Enterprise DBMS erzeugten Audit-Datensätze nutzen.“

S56 **„Redis Enterprise DBMS muss eine zentrale Konfiguration des Inhalts bieten, der in den von allen Komponenten von Redis Enterprise DBMS erzeugten Audit-Records erfasst werden soll.“**

S57 **„Redis Enterprise DBMS muss die Speicherkapazität für Audit-Datensätze in Übereinstimmung mit den vom Unternehmen definierten Speicheranforderungen für Audit-Datensätze zuweisen.“**

S58 **„Redis Enterprise DBMS muss Audit-Daten in eine separate Log-Management-Einrichtung auslagern. Dies muss kontinuierlich und nahezu in Echtzeit für Systeme mit einer Netzwerkverbindung zu der Speichereinrichtung und wöchentlich oder öfter für Stand-Alone-Systeme erfolgen.“**

S59 **„Redis Enterprise DBMS muss eine Warnung an das zuständige Support-Personal ausgeben, wenn das zugewiesene Speichervolumen für Audit-Daten 75 Prozent der maximalen Speicherkapazität für Audit-Daten erreicht.“**

S60 **„Redis Enterprise DBMS muss eine sofortige Echtzeit-Warnung an das zuständige Support-Personal über alle Audit-Protokollausfälle ausgeben.“**

S61 **„Redis Enterprise DBMS muss standardmäßig bei Audit-Fehlern heruntergefahren werden, einschließlich der Nichtverfügbarkeit von Speicherplatz für weitere Audit-Datensätze oder es muss so konfiguriert werden können, dass es bei Audit-Fehlern heruntergefahren wird.“**

S62 **„Redis Enterprise DBMS muss so konfiguriert werden können, dass die ältesten Audit-Protokollsätze zuerst überschrieben werden (FIFO), wenn kein Platz für weitere Audit-Protokollsätze verfügbar ist.“**

S63 **„Redis Enterprise DBMS muss Zeitstempel in Audit-Datensätzen und Anwendungsdaten aufzeichnen, die auf die koordinierte Weltzeit (UTC, früher GMT) abgebildet werden können.“**

S64 **„Die von Redis Enterprise DBMS erzeugten Audit-Informationen müssen vor unberechtigtem Lesezugriff geschützt werden.“**

S65 **„Die von Redis Enterprise DBMS erzeugten Audit-Informationen müssen vor unbefugter Änderung geschützt werden.“**

S66 **„Die von Redis Enterprise DBMS erstellten Audit-Informationen**

müssen vor unbefugtem Löschen geschützt werden.“

S67 **„Redis Enterprise DBMS muss seine Audit-Funktionen vor unberechtigtem Zugriff schützen.“**

S68 **„Redis Enterprise DBMS muss seine Audit-Konfiguration vor unbefugten Änderungen schützen.“**

S69 **„Redis Enterprise DBMS muss seine Audit-Funktionen vor unbefugtem Entfernen schützen.“**

S70 **„Redis Enterprise DBMS muss die Installation von Logikmodulen (Stored Procedures, Funktionen, Triggers, Views, etc.) durch Benutzer ohne expliziten Privilegierungsstatus verbieten.“**

S71 **„Redis Enterprise DBMS muss Zugriffsbeschränkungen im Zusammenhang mit Änderungen an der Konfiguration von Redis Enterprise DBMS oder der Datenbank(en) durchsetzen.“**

S72 **„Redis Enterprise DBMS muss die Berechtigungen zum Ändern von Softwaremodulen einschränken. Dazu gehören gespeicherte Prozeduren, Funktionen und Trigger sowie Links zu Software außerhalb von Redis Enterprise DBMS.“**

S73 **„Das Redis Enterprise DBMS-Software-Installationskonto muss auf autorisierte Benutzer beschränkt sein.“**

S74 **„Die Datenbanksoftware, einschließlich der DBMS-Konfigurationsdateien, muss in speziellen Verzeichnissen oder DASD-Pools gespeichert werden, die vom Host-Betriebssystem und anderen Anwendungen getrennt sind.“**

S75 **„Die Rolle(n)/Gruppe(n), die zur Änderung der Datenbankstruktur (einschließlich, aber nicht notwendigerweise beschränkt auf Tabellen, Indizes, Speicher usw.) und der Logikmodule (gespeicherte Prozeduren, Funktionen, Trigger, Links zu Software außerhalb von Redis Enterprise DBMS usw.) verwendet werden, müssen auf autorisierte Benutzer beschränkt sein.“**

S76 **„Redis Enterprise DBMS muss in Übereinstimmung mit den Sicherheitskonfigurationseinstellungen konfiguriert werden, die auf den DoD-Sicherheitskonfigurations- und Implementierungsrichtlinien basieren, einschließlich STIGs, NSA-Konfigurationsleitfäden, CTOs, DTMs und IAVMs.“**

S77 **„Redis Enterprise DBMS muss Netzwerkfunktionen, Ports, Protokolle und Dienste deaktivieren, die von der Organisation als unsicher eingestuft werden, in Übereinstimmung mit den PPSM-Richtlinien**

(Ports, Protocols, and Services Management).“

S78 **„Redis Enterprise-Produkte müssen eine vom Hersteller unterstützte Version aufweisen.“**

S79 **„Nicht verwendete Datenbankkomponenten, DBMS-Software und Datenbankobjekte müssen entfernt werden.“**

S80 **„Nicht verwendete Datenbankkomponenten, die in Redis Enterprise DBMS integriert sind und nicht deinstalliert werden können, müssen deaktiviert werden.“**

S81 **„Der Zugriff auf externe ausführbare Dateien muss deaktiviert oder eingeschränkt werden.“**

S82 **„Redis Enterprise DBMS muss so konfiguriert werden, dass die Verwendung von organisationsdefinierten Funktionen, Ports, Protokollen und/oder Diensten, wie in der PPSM CAL und den Schwachstellenbewertungen definiert, untersagt oder eingeschränkt wird.“**

S83 **„Redis Enterprise DBMS muss Benutzer dazu auffordern, sich erneut zu authentifizieren, wenn vom Unternehmen definierte Umstände oder Situationen eine erneute Authentifizierung erfordern.“**

S84 **„Redis Enterprise DBMS muss organisatorische Benutzer (oder Prozesse, die im Namen von organisatorischen Benutzern handeln) eindeutig identifizieren und authentifizieren.“**

S85 **„Wenn Passwörter zur Authentifizierung verwendet werden, darf Redis Enterprise DBMS nur gehashte, gesalzene Repräsentationen von Passwörtern speichern.“**

S86 **„Redis Enterprise DBMS muss die Verwendung von zwischengespeicherten Authentifikatoren nach einer von der Organisation festgelegten Zeitspanne untersagen.“**

S87 **„Redis Enterprise DBMS muss, wenn es PKI-basierte Authentifizierung verwendet, Zertifikate validieren, indem es eine RFC 5280-konforme Validierung des Zertifizierungspfads durchführt.“**

S88 **„Redis Enterprise DBMS muss einen autorisierten Zugriff auf alle von Redis Enterprise DBMS gespeicherten/verwendeten privaten PKI-Schlüssel erzwingen.“**

× **„Redis Enterprise DBMS muss die PKI-authentifizierte Identität einem zugehörigen Benutzerkonto zuordnen.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S89 **„Redis Enterprise DBMS muss die Rückmeldung von Authentifizie-**

rungsinformationen während des Authentifizierungsprozesses verbergen, um die Informationen vor einer möglichen Ausnutzung/-Verwendung durch nicht autorisierte Personen zu schützen.“

S90 **„Redis Enterprise DBMS muss NIST FIPS 140-2-validierte kryptografische Module für kryptografische Operationen verwenden.“**

S91 **„Redis Enterprise DBMS muss nicht-organisatorische Benutzer (oder Prozesse, die im Namen von nicht-organisatorischen Benutzern handeln) eindeutig identifizieren und authentifizieren.“**

S92 **„Redis Enterprise DBMS muss NSA-zugelassene Kryptografie verwenden, um Verschlusssachen in Übereinstimmung mit den Anforderungen der Dateneigentümer zu schützen.“**

S93 **„Redis Enterprise DBMS muss NIST FIPS 140-2-validierte kryptografische Module implementieren, um digitale Signaturen bereitzustellen.“**

S94 **„Redis Enterprise DBMS muss NIST FIPS 140-2-validierte kryptografische Module implementieren, um kryptografische Hashes zu erzeugen und zu validieren.“**

S95 **„Redis Enterprise DBMS muss NIST FIPS 140-2-validierte kryptografische Module implementieren, um nicht klassifizierte Informationen, die Vertraulichkeit und kryptografischen Schutz erfordern, in Übereinstimmung mit den Anforderungen der Dateneigentümer zu schützen.“**

S96 **„Redis Enterprise DBMS muss die Benutzerfunktionalität (einschließlich Benutzerschnittstellendienste) von der Datenbankmanagementfunktionalität trennen.“**

S97 **„Der Zugriff auf die Redis Enterprise-Kontrollebene muss eingeschränkt sein.“**

S98 **„Redis Enterprise DBMS darf nur systemgenerierte Sitzungsbezeichner erkennen.“**

S99 **„Redis Enterprise DBMS muss die Authentizität von Kommunikationssitzungen aufrechterhalten, indem es vor MITM-Angriffen schützt, die die Sitzungskennungen erraten.“**

S100 **„Redis Enterprise DBMS darf nur Endteilnehmerzertifikate akzeptieren, die von DoD PKI oder DoD-zugelassenen PKI-Zertifizierungsstellen (CAs) für die Einrichtung aller verschlüsselten Sitzungen ausgestellt wurden.“**

S101 **„Redis Enterprise DBMS muss in einen sicheren Zustand überge-**

hen, wenn die Systeminitialisierung fehlschlägt, das Herunterfahren fehlschlägt oder Abbrüche fehlschlagen.“

S102 **„Im Falle eines Systemausfalls muss Redis Enterprise DBMS alle Informationen aufbewahren, die zur Ermittlung der Ausfallursache und zur Wiederaufnahme des Betriebs mit möglichst geringer Unterbrechung der Missionsprozesse erforderlich sind.“**

S103 **„Redis Enterprise DBMS muss die Vertraulichkeit und Integrität aller Informationen im laufenden Betrieb schützen.“**

S104 **„Redis Enterprise DBMS muss kryptografische Mechanismen implementieren, um eine unbefugte Änderung der von der Organisation definierten ruhenden Informationen (die mindestens PII und klassifizierte Informationen umfassen) auf von der Organisation definierten Informationssystemkomponenten zu verhindern.“**

S105 **„Redis Enterprise DBMS muss kryptografische Mechanismen implementieren, die die unbefugte Offenlegung von organisationsdefinierten Informationen im Ruhezustand auf organisationsdefinierten Informationssystemkomponenten verhindern.“**

× **„Datenbankinhalte müssen durch die Durchsetzung einer Datenübertragungsrichtlinie vor unbefugter und unbeabsichtigter Informationsübertragung geschützt werden.“**

Entfällt, da dies durch organisatorische, prozessuale Maßnahmen umgesetzt werden muss.

S106 **„Redis Enterprise DBMS muss die unbefugte und unbeabsichtigte Übertragung von Informationen über gemeinsam genutzte Systemressourcen verhindern.“**

S107 **„Der Zugriff auf Datenbankdateien muss auf relevante Prozesse und autorisierte, administrative Benutzer beschränkt sein.“**

S108 **„Redis Enterprise DBMS muss die Vertraulichkeit und Integrität von Informationen während der Vorbereitung zur Übertragung wahren.“**

S109 **„Redis Enterprise DBMS muss die Vertraulichkeit und Integrität von Informationen während des Empfangs sicherstellen.“**

S110 **„Redis Enterprise DBMS und zugehörige Anwendungen müssen die Verwendung der dynamischen Codeausführung für Situationen reservieren, die dies erfordern.“**

S111 **„Redis Enterprise DBMS und zugehörige Anwendungen müssen bei der Verwendung der dynamischen Codeausführung die Eingabeda-**

ten auf ungültige Werte prüfen, die auf einen Code-Injection-Angriff hindeuten könnten.“

S112 **„Wenn Updates auf die Redis Enterprise DBMS-Software angewendet werden, müssen alle Softwarekomponenten, die ersetzt oder überflüssig gemacht wurden, entfernt werden.“**

S113 **„Sicherheitsrelevante Software-Updates für Redis Enterprise DBMS müssen innerhalb des von einer maßgeblichen Quelle (z. B. IAVM, CTOs, DTMs und STIGs) vorgegebenen Zeitraums installiert werden.“**

S114 **„Redis Enterprise DBMS muss Audit-Datensätze für DoD-definierte auditierbare Ereignisse innerhalb aller DBMS/Datenbank-Komponenten erzeugen.“**

S115 **„Wenn die DBMS-Authentifizierung mit Passwörtern erfolgt, muss Redis Enterprise DBMS die DoD-Standards für Passwortkomplexität und -lebensdauer durchsetzen.“**

Literaturverzeichnis

[1] Bitkom e. V. *203 Milliarden Euro Schaden pro Jahr durch Angriffe auf deutsche Unternehmen*. [Online; Zugriff am 3. September 2022]. 2022. URL: https://www.bitkom.org/Presse/Presseinformation/Wirtschaftsschutz-2022.

[2] Bitkom e. V. *Angriffsziel deutsche Wirtschaft: mehr als 220 Milliarden Euro Schaden pro Jahr*. [Online; Zugriff am 3. September 2022]. 2021. URL: https://www.bitkom.org/Presse/Presseinformation/Angriffsziel-deutsche-Wirtschaft-mehr-als-220-Milliarden-Euro-Schaden-pro-Jahr.

[3] Bitkom e. V. *Welche der folgenden Arten von digitalen IT-Angriffen haben innerhalb der letzten 12 Monate in Ihrem Unternehmen einen Schaden verursacht?* [Online; Zugriff am 3. September 2022]. 2022. URL: https://de.statista.com/statistik/daten/studie/928943/umfrage/von-digitalen-angriffen-betroffene-unternehmen-nach-art-des-angriffs/.

[4] Bundeskriminalamt. *Bundeslagebild Cybercrime 2020*. [Online; Zugriff am 7. Juli 2022]. 2021. URL: https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/JahresberichteUndLagebilder/Cybercrime/cybercrimeBundeslagebild2020.html.

[5] Bundeskriminalamt. *Cybercrime*. [Online; Zugriff am 12. Juli 2022]. 2021. URL: https://www.bka.de/DE/UnsereAufgaben/Deliktsbereiche/Cybercrime/cybercrime_node.html.

[6] Bundeskriminalamt. *Bundeslagebild Cybercrime 2021*. [Online; Zugriff am 9. Juli 2022]. 2022. URL: https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/JahresberichteUndLagebilder/Cybercrime/cybercrimeBundeslagebild2021.html.

[7] Lutz Bellmann u. a. „Digitalisierungsschub in Firmen während der Corona-Pandemie“. In: *Wirtschaftsdienst* 101.9 (2021). DOI: 10.1007/s10273-021-3005-3.

[8] Heiner Lasi u. a. „Industrie 4.0“. In: *Wirtschaftsinformatik* 56.4 (2014), S. 261–264. DOI: 10.1007/s11576-014-0424-4.

[9] Bundesamt für Sicherheit in der Informationstechnik. *Die Lage der IT-Sicherheit in Deutschland 2022*. [Online; Zugriff am 3. November 2022]. 2022. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2022.pdf?__blob=publicationFile&v=6.

[10] MongoDB Inc. *NoSQL vs. SQL Databases*. [Online; Zugriff am 14. November 2022]. 2022. URL: https://www.mongodb.com/nosql-explained/nosql-vs-sql.

[11] ABM Moniruzzaman und Syed Akhter Hossain. „Nosql database: New era of databases for big data analytics-classification, characteristics and comparison“. In: *arXiv preprint arXiv:1307.0191* (2013).

[12] Bundesamt für Sicherheit in der Informationstechnik. *Leitfaden IT-Sicherheit*. [Online; Zugriff am 23. September 2022]. 2007. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Leitfaden/GS-Leitfaden_pdf.pdf.

[13] Supriya Saxena. *Difference between Database and DBMS*. [Online; Zugriff am 8. Juli 2022]. 2022. URL: https://www.geeksforgeeks.org/difference-between-database-and-dbms/.

[14] IBM Corporation. *What is a database management system?* [Online; Zugriff am 10. Juli 2022]. 2022. URL: https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system.

[15] Thomas M. Connolly und Carolyn E. Begg. *Database Systems - A Practical Approach to Design Implementation and Management*. 6. Auflage. Pearson Verlag, 2014. ISBN: 978-1292061184.

[16] Sabrina Sicari, Alessandra Rizzardi und Alberto Coen-Porisini. „Security & privacy issues and challenges in NoSQL databases“. In: *Computer Networks* 206 (2022). DOI: 10.1016/j.comnet.2022.108828.

[17] Statista Research Department. *Ranking of the most popular database management systems worldwide, as of January 2022*. [Online; Zugriff am 5. Juni 2022]. 2022. URL: https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/.

[18] Stefan Luber und Nico Litzel. *Was ist eine relationale Datenbank?* [Online; Zugriff am 5. August 2022]. 2017. URL: https://www.bigdata-insider.de/was-ist-eine-relationale-datenbank-a-643028/.

[19] Tutorials Point India Private Limited. *Relation Data Model*. [Online; Zugriff am 5. August 2022]. 2022. URL: https://www.tutorialspoint.com/dbms/relational_data_model.htm.

[20] Refsnes Data und W3schools Network. *SQL Constraints*. [Online; Zugriff am 5. August 2022]. 2022. URL: https://www.w3schools.com/sql/sql_constraints.asp.

[21] Refsnes Data und W3schools Network. *SQL NOT NULL Constraint*. [Online; Zugriff am 5. August 2022]. 2022. URL: https://www.w3schools.com/sql/sql_notnull.asp.

[22] Refsnes Data und W3schools Network. *SQL UNIQUE Constraint*. [Online; Zugriff am 5. August 2022]. 2022. URL: https://www.w3schools.com/sql/sql_unique.asp.

[23] Refsnes Data und W3schools Network. *SQL FOREIGN KEY Constraint*. [Online; Zugriff am 5. August 2022]. 2022. URL: https://www.w3schools.com/sql/sql_foreignkey.asp.

[24] Troy Hunt. *Have I Been Pwned API v3*. [Online; Zugriff am 8. August 2022]. 2022. URL: https://haveibeenpwned.com/api/v3/breaches.

[25] *What is a Parent table and a Child table in Database?* [Online; Zugriff am 12. August 2022]. 2016. URL: https://stackoverflow.com/a/35187154.

[26] Simran Srivastava. *Advantages and Disadvantages of SQL*. [Online; Zugriff am 15. August 2022]. 2021. URL: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-sql/.

[27] GridGain Systems Inc. *5 Limitations of MySQL with Big Data*. [Online; Zugriff am 15. August 2022]. 2018. URL: https://www.gridgain.com/resources/blog/5-limitations-mysql-big-data.

[28] Stefan Luber und Nico Litzel. *Was ist NoSQL?* [Online; Zugriff am 18. August 2022]. 2017. URL: https://www.bigdata-insider.de/was-ist-nosql-a-615718/.

[29] DB-Engines.com. *DBMS popularity broken down by database model*. [Online; Zugriff am 18. August 2022]. 2022. URL: https://db-engines.com/en/ranking_categories.

[30] MongoDB Inc. *Understanding the Different Types of NoSQL Databases*. [Online; Zugriff am 20. August 2022]. 2022. URL: https://www.mongodb.com/scale/types-of-nosql-databases.

[31] Jon Erik Solheim. *Object relations in a NoSQL database*. [Online; Zugriff am 19. August 2022]. 2017. URL: https://restdb.io/blog/object-relations-in-a-nosql-database.

[32] DB-Engines.com. *DB-Engines Ranking of Document Stores*. [Online; Zugriff am 19. August 2022]. 2022. URL: https://db-engines.com/en/ranking/document+store.

[33] Redis Ltd. *Redis data types tutorial*. [Online; Zugriff am 20. August 2022]. 2022. URL: https://redis.io/docs/data-types/tutorial/.

[34] DB-Engines.com. *DB-Engines Ranking of Key-value Stores*. [Online; Zugriff am 19. August 2022]. 2022. URL: https://db-engines.com/en/ranking/key-value+store.

[35] MongoDB Inc. *Types Of NoSQL Database Management Systems*. [Online; Zugriff am 21. August 2022]. 2022. URL: https://www.mongodb.com/scale/types-of-nosql-database-management-systems.

[36] Neo4j Inc. *What is a Graph Database?* [Online; Zugriff am 22. August 2022]. 2022. URL: https://neo4j.com/developer/graph-database/.

[37] DB-Engines.com. *DB-Engines Ranking of Graph DBMS*. [Online; Zugriff am 20. August 2022]. 2022. URL: https://db-engines.com/en/ranking/graph+dbms.

[38] Amazon Web Services Inc. *What is a Columnar Database?* [Online; Zugriff am 22. August 2022]. 2022. URL: https://aws.amazon.com/nosql/columnar/.

[39] Emrah Idman. *Compression Techniques for Column Oriented Databases*. [Online; Zugriff am 22. August 2022]. 2022. URL: https://neo4j.com/developer/graph-database/.

[40] DB-Engines.com. *DB-Engines Ranking of Wide Column Stores*. [Online; Zugriff am 21. August 2022]. 2022. URL: https://db-engines.com/en/ranking/wide+column+store.

[41] Bundesamt für Sicherheit in der Informationstechnik. *Umsetzungshinweise zum Baustein ISMS.1 Sicherheitsmanagement*. [Online; Zugriff am 26. August 2022]. 2021. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Umsetzungshinweise/Umsetzungshinweise_2022/Umsetzungshinweis_zum_Baustein_ISMS_1_Sicherheitsmanagement.pdf.

[42] DevSec Hardening Framework Team. *DevSec Project*. [Online; Zugriff am 27. August 2022]. 2018. URL: https://dev-sec.io/project/.

[43] Bundesamt für Sicherheit in der Informationstechnik. *Historie des BSI*. [Online; Zugriff am 27. Juli 2022]. 2022. URL: https://www.bsi.bund.de/DE/Das-BSI/BSI-Historie/bsi-historie_node.html.

[44] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Zertifizierung von ausgelagerten Komponenten*. [Online; Zugriff am 27. Juli 2022]. 2004. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Zertifikat/Veroeffentl/Outsourcing_pdf.html.

[45] Bundesamt für Sicherheit in der Informationstechnik. *Edition 2018 des IT-Grundschutz-Kompendiums*. [Online; Zugriff am 28. Juli 2022]. 2018. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2018.pdf.

[46] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standards*. [Online; Zugriff am 28. Juli 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/BSI-Standards/bsi-standards_node.html.

[47] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standard 200-4: Business Continuity Management (Community Draft)*. [Online; Zugriff am 27. Juli 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/BSI-Standards/BSI-Standard-200-4-Business-Continuity-Management/bsi-standard-200-4_Business_Continuity_Management_node.html.

[48] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standard 200-4: Hilfsmittel*. [Online; Zugriff am 27. August 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/BSI-Standards/BSI-Standard-200-4-Business-Continuity-Management/BSI-Standard-200-4_Hilfsmittel/BSI_Standard_200_4_Hilfsmittel_node.html.

[49] Bundesamt für Sicherheit in der Informationstechnik. *Technische Richtlinien*. [Online; Zugriff am 27. Juli 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/technische-richtlinien_node.html.

[50] Bundesamt für Sicherheit in der Informationstechnik. *Liste der Technische Richtlinien und Anhänge aufgelistet nach Änderungsdatum*. [Online; Zugriff am 18. November 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/Liste-TR-nach-Aenderungsdatum/traenderungen_node.html.

[51] Center for Internet Security Inc. *CIS: Celebrating 20 Years of Cybersecurity*. [Online; Zugriff am 2. August 2022]. 2022. URL: https://www.cisecurity.org/insights/blog/cis-celebrating-20-years-of-cybersecurity.

[52] Center for Internet Security Inc. *About us*. [Online; Zugriff am 4. Oktober 2022]. 2022. URL: https://www.cisecurity.org/about-us.

[53] Center for Internet Security Inc. *20 Years of Creating Confidence in the Connected World*. [Online; Zugriff am 2. August 2022]. 2022. URL: https://www.cisecurity.org/insights/blog/20-years-of-creating-confidence-in-the-connected-world.

[54] Clint Kreitner und Bert Miuccio. *The Center for Internet Security: Global Security Benchmarks for Computers Connected to the Internet*. [Online; Zugriff am 2. August 2022]. 2014. URL: https://web.archive.org/web/20140312224452/http://www.isaca.org/Journal/Past-Issues/2001/Volume-6/Pages/The-Center-for-Internet-Security-Global-Security-Benchmarks-for-Computers-Connected-to-the-Internet.aspx.

[55] Center for Internet Security Inc. *The 18 CIS Critical Security Controls*. [Online; Zugriff am 2. August 2022]. 2022. URL: https://www.cisecurity.org/controls/cis-controls-list.

[56] Center for Internet Security Inc. *CIS Critical Security Controls FAQ*. [Online; Zugriff am 2. August 2022]. 2022. URL: https://www.cisecurity.org/controls/cis-controls-faq.

[57] National Institute of Standards und Technology. *SP 800-53 Rev. 5, Security and Privacy Controls for Information Systems and Organizations*. [Online; Zugriff am 8. August 2022]. 2020. URL: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final.

[58] National Institute of Standards und Technology. *Cybersecurity*. [Online; Zugriff am 9. August 2022]. 2022. URL: https://www.nist.gov/cybersecurity.

[59] LLC SolarWinds Worldwide. *Understanding DISA STIG Compliance Requirements*. [Online; Zugriff am 15. August 2022]. 2015. URL: https://www.solarwinds.com/federal-government/solution/disa-stig-compliance.

[60] Defense Information Systems Agency. *STIGs Home*. [Online; Zugriff am 9. August 2022]. 2015. URL: https://web.archive.org/web/20150122135956/https://iase.disa.mil/stigs/Pages/index.aspx.

[61] National Institute of Standards und Technology. *IA-enabled product - Glossary / CSRC*. [Online; Zugriff am 11. August 2022]. 2022. URL: https://csrc.nist.gov/glossary/term/IA_enabled_product.

[62] Center for Internet Security Inc. *CIS Cloud Security Resources for STIG Compliance*. [Online; Zugriff am 2. August 2022]. 2022. URL: https://www.cisecurity.org/insights/blog/new-options-from-cis-for-stig-compliance.

[63] Center for Internet Security Inc. *How to Meet STIG Compliance and Achieve OS Security with CIS*. [Online; Zugriff am 9. August 2022]. 2022. URL: https://www.cisecurity.org/insights/blog/how-to-meet-stig-compliance-and-achieve-os-security-with-cis.

[64] Defense Information Systems Agency. *STIGs Document Library*. [Online; Zugriff am 9. August 2022]. 2022. URL: https://public.cyber.mil/stigs/downloads/.

[65] Inc Opscode. *Opscode Announces $2.5 Million in Series A Round*. [Online; Zugriff am 11. August 2022]. 2009. URL: https://www.chef.io/blog/opscode-announces-2-5-million-in-series-a-round.

[66] Beth Pariseau. *Progress steers Chef InSpec toward CSPM*. [Online; Zugriff am 1. September 2022]. 2021. URL: https://www.techtarget.com/searchitoperations/news/252506701/Progress-steers-Chef-InSpec-toward-CSPM.

[67] Nick Rycar. *Introducing Chef Infra Client 16*. [Online; Zugriff am 1. September 2022]. 2020. URL: https://www.chef.io/blog/introducing-chef-infra-client-16.

[68] Progress Software Corporation. *Industry Recognized Policy-based Compliance Automation Software Tools*. [Online; Zugriff am 2. September 2022]. 2022. URL: https://www.chef.io/products/chef-inspec.

[69] Progress Software Corporation. *An Overview of Chef InSpec*. [Online; Zugriff am 2. September 2022]. 2022. URL: https://docs.chef.io/inspec/.

[70] Progress Software Corporation. *About Chef InSpec Profiles*. [Online; Zugriff am 2. September 2022]. 2022. URL: https://docs.chef.io/inspec/profiles/.

[71] Edmund Haselwanter u. a. *DevSec MySQL Baseline*. [Online; Zugriff am 2. September 2022]. 2022. URL: https://github.com/dev-sec/mysql-baseline.

[72] Progress Software Corporation. *Chef InSpec Universal Matchers Reference*. [Online; Zugriff am 6. September 2022]. 2022. URL: https://docs.chef.io/inspec/matchers/.

[73] *Is there any difference between apache2 and httpd?* [Online; Zugriff am 6. September 2022]. 2016. URL: https://askubuntu.com/a/600902.

[74] CIS Center for Internet Security. *CIS Oracle MySQL Enterprise Edition 8.0 Benchmark v1.2.0*. [Online; Zugriff am 24. Juni 2022]. 2022. URL: https://www.cisecurity.org/benchmark/oracle_mysql.

[75] Progress Software Corporation. *command resource*. [Online; Zugriff am 9. September 2022]. 2022. URL: https://docs.chef.io/inspec/resources/command/.

[76] Progress Software Corporation. *parse_config_file resource*. [Online; Zugriff am 9. September 2022]. 2022. URL: https://docs.chef.io/inspec/resources/parse_config_file/.

[77] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standard 200-2 - IT-Grundschutz-Methodik*. [Online; Zugriff am 14. September 2022]. 2017. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/BSI_Standards/standard_200_2.pdf.

[78] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Bausteine - Edition 2022*. [Online; Zugriff am 27. November 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/IT-Grundschutz-Bausteine/Bausteine_Download_Edition_node.html.

[79] CIS Center for Internet Security. *CIS Apache Cassandra 3.11 Benchmark v1.0.0*. [Online; Zugriff am 22. August 2022]. 2019. URL: https://www.cisecurity.org/benchmark/apache_cassandra.

[80] CIS Center for Internet Security. *CIS MongoDB 5 Benchmark v1.1.0*. [Online; Zugriff am 22. August 2022]. 2022. URL: https://www.cisecurity.org/benchmark/mongodb.

[81] CIS Center for Internet Security. *CIS PostgreSQL 14 Benchmark v1.0.0*. [Online; Zugriff am 22. August 2022]. 2021. URL: https://www.cisecurity.org/benchmark/postgresql.

[82] Defense Information Systems Agency. *MongoDB Enterprise Advanced 4.x STIG Version 1, Release 1*. [Online; Zugriff am 22. August 2022]. 2022. URL: https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_MDB_Enterprise_Advanced_4-x_V1R1_STIG.zip.

[83] Defense Information Systems Agency. *Redis Enterprise 6.x STIG Version 1, Release 1*. [Online; Zugriff am 22. August 2022]. 2021. URL: https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_Redis_Enterprise_6-x_V1R1_STIG.zip.

[84] National Institute of Standards und Technology. *FIPS PUB 180-4: Secure Hash Standard (SHS)*. [Online; Zugriff am 21. Oktober 2022]. 2015. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[85] National Institute of Standards und Technology. *FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. [Online; Zugriff am 21. Oktober 2022]. 2015. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[86] Neo4j Inc. *Release Notes: Neo4j 5*. [Online; Zugriff am 14. September 2022]. 2022. URL: https://neo4j.com/release-notes/database/neo4j-5/.

[87] Neo4j Inc. *Configuration settings*. [Online; Zugriff am 14. September 2022]. 2022. URL: https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/.

[88] Redis Ltd. *Redis configuration*. [Online; Zugriff am 14. September 2022]. 2022. URL: https://redis.io/docs/management/config/.

[89] Redis Ltd. *AUTH*. [Online; Zugriff am 9. November 2022]. 2022. URL: https://redis.io/commands/auth/.

[90] Neo4j Inc. *Built-in roles*. [Online; Zugriff am 12. November 2022]. 2022. URL: https://neo4j.com/docs/operations-manual/5/authentication-authorization/built-in-roles/.

[91] Patrick Münch u. a. *DevSec Linux Baseline - package-08*. [Online; Zugriff am 15. November 2022]. 2022. URL: https://github.com/dev-sec/linux-baseline/blob/666e7092534bc29554700c21c6b8864cbc45eeae/controls/package_spec.rb#L78.

[92] Oracle Corporation. *JPDA Connection and Invocation*. [Online; Zugriff am 1. November 2022]. 2011. URL: https://docs.oracle.com/javase/7/docs/technotes/guides/jpda/conninv.html.

[93] Patrick Pelletier. *Diffie-Hellman parameters*. [Online; Zugriff am 20. November 2022]. 2021. URL: https://wiki.openssl.org/index.php?title=Diffie-Hellman_parameters&oldid=3178.

[94] Bundesamt für Sicherheit in der Informationstechnik. *Lerneinheit 6.3: Dokumentation*. [Online; Zugriff am 23. November 2022]. 2022. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/Zertifizierte-Informationssicherheit/IT-Grundschutzschulung/Online-Kurs-IT-Grundschutz/Lektion_6_IT-Grundschutz-Check/Lektion_6_03/Lektion_6_03_node.html.

[95] Titania Ltd. *DISA STIG Compliance Explained*. [Online; Zugriff am 24. November 2022]. 2022. URL: https://www.titania.com/resources/guides/disa-stig-compliance-explained/.

[96] Progress Software Corporation. *Chef InSpec Language*. [Online; Zugriff am 26. November 2022]. 2022. URL: https://docs.chef.io/inspec/dsl_inspec/.

[97] Nancy R. Tague. *Quality Toolbox*. 2. Auflage. ASQ Quality Press, 2005. ISBN: 978-0873896399.

[98] Artur Sosin. „How to increase the Information Assurance in the Information Age“. In: *Journal of Defense Resources Management* 9 (2018), S. 45–57.

[99] DKE - VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V. *Grundlagen der Normung*. [Online; Zugriff am 28. November 2022]. 2022. URL: https://www.dke.de/de/normen-standards/grundlagen-der-normung.

Bildverzeichnis

Tabellenverzeichnis

Listingverzeichnis

Abkürzungsverzeichnis

ACL	Access Control List. 43, 46, 47, 59, 74, 75, 83, 88–94, 103, 112, 118, 120, 121, 136
AES	Advanced Encryption Standard. 63–65
API	Programmierschnittstelle. 8, 10, 61, 136, 143, 147, 148, 150, 155, 159
APOC	Awesome Procedures On Cypher. 111, 117
APT	Advanced Packaging Tool. 31, 32, 78–81
BCM	Business Continuity Management. 19
BKA	Bundeskriminalamt. 5
BSI	Bundesamt für Sicherheit in der Informationstechnik. 3, 6, 17–19, 26–34, 36–39, 43–53, 57, 59–61, 63–69, 71, 73, 74, 137, 140, 162
BSI-TR	Technische Richtlinien des BSI. 17, 19, 28, 50, 51, 62–65
CBC	Cipher Block Chaining. 64
CC	Common Criteria. 19
CIS	Center for Internet Security. 2, 3, 18–21, 26–28, 31, 34, 35, 41, 44–48, 53–63, 65–74, 137, 138, 140, 182
CVSS	Common Vulnerability Scoring System. 142
DBMS	Datenbankmanagementsystem. 8, 9, 12, 24, 29–31, 34–76, 124, 138, 163, 165, 170–172, 174–176, 190, 192, 195–201
DH	Diffie-Hellman. 105, 106
DHE	Diffie-Hellman Ephemeral. 63, 64
DISA	Defense Information Systems Agency. 18, 21, 140
DNF	Dandified Yum. 31
DoD	United States Department of Defense. 21, 137, 190, 192, 194, 195, 197, 199, 201
DPKG	Debian Package Management System. 31
DSL	Domain Specific Language. 23, 24
ECC	Elliptische-Kurven-Kryptographie. 19
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral. 63, 64
ECDSA	Elliptic Curve Digital Signature Algorithm. 64
FIFO	First In – First Out. 56, 196
FIPS	Federal Information Processing Standard. 184, 188, 191, 199
FISMA	Federal Information Security Modernization Act. 21
FQDN	Fully Qualified Domain Name. 48

GDS Graph Data Science. 111

IAM Identity and Access Management. 35–39, 41, 44, 48–50, 52, 53, 162
IANA Internet Assigned Numbers Authority. 72
IdM Identity Management. 162
IDS Intrusion Detection System. 53
IPS Intrusion Prevention System. 53
ISMS Information Security Management System. 17, 18

JDWP Java Debug Wire Protocol. 102

LDAP Lightweight Directory Access Protocol. 35, 41, 83
LZ4 Lempel-Ziv-Vier. 16
LZO Lempel-Ziv-Oberhumer. 16

MAC Message Authentication Code. 64
MITM Man-in-the-Middle. 192, 199

NIST National Institute of Standards and Technology. 21, 191, 199
NTP Network Time Protocol. 74, 116

PDCA Plan–Do–Check–Act. 206
PFS Perfect Forward Secrecy. 63, 105
PoLP Principle of Least Privilege. 47

RLS Row Level Security. 68, 187
RPM Red Hat Package Manager. 31
RSA Rivest-Shamir-Adleman. 64

SASL Simple Authentication and Security Layer. 41
SAST Static Application Security Testing. 33
SIEM Security Information and Event Management. 53–56, 164
SNMP Simple Network Management Protocol. 60, 102
SQL Structured Query Language. 5, 6, 9, 12, 13, 17, 24, 68, 141, 172, 187
STIGs Security Technical Implementation Guides. 2, 3, 18, 21, 26, 28–30, 33, 34, 36–41, 43, 46, 47, 50, 53–61, 63–70, 73, 136–138, 140, 190, 194, 197, 201

YUM Yellowdog Updater, Modified. 31

Glossar

Demingkreis	Der Demingkreis oder auch PDCA-Zyklus beschreibt einen iterativen Vier-Phasen-Prozess zur Überwachung und fortwährenden Verbesserung von Prozessen oder Produkten [97]. 28, 123, 138, 141
IA	Information Assurance (deutsch Informationssicherung, abgekürzt IA) ist die Praxis der Sicherung von Informationen und des Risikomanagements im Zusammenhang mit der Nutzung, Verarbeitung, Speicherung und Übertragung von Informationen. Informationssicherung umfasst den Schutz der Integrität, Verfügbarkeit, Authentizität, Nichtabstreitbarkeit und Vertraulichkeit von Nutzerdaten [98]. 21
IEC	Die Internationale Elektrotechnische Kommission (IEC) ist eine internationale Normungsorganisation, die internationale Normen im Bereich der Elektrotechnik ausarbeitet und veröffentlicht [99]. 18
ISO	Die Internationale Organisation für Normung (ISO) ist die internationale Vereinigung von Normungsorganisationen, die internationale Normen in allen Bereichen mit Ausnahme der Elektrotechnik und Telekommunikation ausarbeitet und veröffentlicht [99]. 18
NoSQL	"Not only SQL" oder, inzwischen seltener, "non-SQL" oder auch "non-relational" umfasst mehrere alternative Ansätze zu relationalen Datenbankmodellen. 6, 12, 13, 16, 17, 138, 141

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.
Die Stellen der Arbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Ich erkläre ferner, dass ich die vorliegende Arbeit in keinem anderen Prüfungsverfahren als Prüfungsarbeit eingereicht habe oder einreichen werde.

Die eingereichte schriftliche Fassung entspricht der auf dem Medium gespeicherten Fassung.

Bremen, 28. November 2022

Ort, Datum Unterschrift

Thesen

Master-Thesis
Konzeption eines Vorgehensmodells zur Ableitung von Härtungsmaßnahmen für nicht-relationale Datenbanksysteme

Eingereicht am:	28. November 2022
von:	Lukas Zorn
Betreuerin:	Prof. Dr.-Ing. Antje Raab-Düsterhöft
Zweitbetreuer:	Prof. Dr.-Ing. habil. Andreas Ahrens

- Das entwickelte Vorgehensmodell bietet eine gute Arbeitsgrundlage für die Ableitung technischer Härtungsmaßnahmen für nicht-relationale Datenbanksysteme.
- Zur Identifizierung aller sicherheitsrelevanten Konfigurationsoptionen ist eine strukturierte, zielführende Vorgehensweise erforderlich.
- Das Vorgehensmodell muss als fortlaufender PDCA-Zyklus verstanden und entsprechend weiterentwickelt werden.
- Steht für ein Datenbanksystem keine Härtungsrichtlinie zur Verfügung, können oftmals Leitlinien anderer Datenbanksysteme zur Orientierung herangezogen werden.
- Die Community-Editionen vieler nicht-relationaler Datenbanksysteme verfügen im Gegensatz zu den entsprechenden Enterprise-Editionen nicht über die für einen sicheren Betrieb erforderlichen grundlegenden Sicherheitsfunktionalitäten.
- Die Community-Edition von Redis war in der Lage, die durch das Vorgehensmodell festgelegten Anforderungen zu $\approx 66,67$ % abzudecken, während Neo4j diese nur zu $\approx 64,20$ % erfüllen konnte.
- Fehlende Sicherheitsfunktionalitäten können in der Regel nicht mit vertretbarem Aufwand und in angemessener Qualität selbständig nachgerüstet werden.
- Zahlreiche nicht-relationale Datenbanksysteme verfügen nicht über eine sichere Standard-Konfiguration.