Hochschule Wismar
Fakultät Ingenieurswissenschaften
Bereich Elektrotechnik und Informatik

Eine Teilanalyse sicherheitsrelevanter Aspekte von Nicht-Relationalen Datenbanken

Wie setzen couchDB unnd rethinkDB Datenbankhärtung analog zu den BSI-Grundschutzbausteinen APP.4.3 und APP.6 um?

Eingereicht am	29. Oktober 2023 Sommersemester 2023
von	Florian Priegnitz Matrikelnr.
Dozentin	Prof. Dr. Antje Raab-Düsterhöft
Modul	IT-Forensik Projekt II

Abstrakt (deutsch)

Die Arbeit fokussiert sich auf die (Teil-)Analyse der Implementierung von Sicherheitsfunktionen in nicht-relationalen Datenbankmanagementsystemen (DBMS), exemplarisch dargelegt an CouchDB und RethinkDB. Grundlage bildet das IT-Grundschutz-Kompendium des Bundesamts für Sicherheit in der Informationstechnik (BSI), aufbereitet durch Kavalir (2021). Aufgrund des Fehlens spezifischer BSI-Leitlinien für nicht-relationale DBMS, diente der Systembaustein "APP.4.3: Relationale Datenbanken" als zentrales Element dieser Untersuchung. Ziel ist es, die Übertragbarkeit der Methodik zu prüfen und Stärken sowie Schwächen zu identifizieren, um eine mögliche Verbesserung des Vorgehens zu evaluieren. Im Vergleich zu Kavalir (2021) stellt diese Analyse eine Teilanalyse dar, da sie nicht den gesamten Informationsverbund abdeckt.

Abstract (english)

The focus of this work is on the (partial) analysis of the implementation of security features in non-relational database management systems (DBMS), illustrated through the examples of CouchDB and RethinkDB. The foundation for this analysis is laid by the IT Baseline Protection Compendium from the Federal Office for Information Security (short BSI), as prepared by Kavalir (2021). Due to the absence of specific BSI guidelines for non-relational DBMS, the system component "APP.4.3: Relational Databases" served as a central element of this examination. The objective is to assess the transferability of the methodology, and to identify strengths and weaknesses to evaluate potential improvements in the approach. Compared to Kavalir (2021), this analysis constitutes a partial examination as it does not encompass the entire information network.

Inhalt

Abstrakt (deutsch) 2
Abstract (english) 2
Abbildungsverzeichnis 5
Tabellenverzeichnis 5
1. Aufgabenstellung 6
2. Grundlagen 7
2.1 Abgrenzung Relationale Datenbanken von Nicht-Relationalen Datenbanken 8
2.1.1 CAP-Theorem 9
2.1.2 ACID 10
2.1.3 BASE 11
2.2 Arten von NoSQL-Datenbanken 11
2.2.1 Dokumentenbasierte Datenbanken 11
2.2.2 Schlüssel-Wert-Datenbanken 13
2.2.3 Spaltenorientierte Datenbanken 14
2.2.4 Graphdatenbanken 14
2.3 Bundesamt für Sicherheit in der Informationstechnik (BSI) 14
2.3.1 IT-Grundschutz-Kompendium des BSI 15
2.3.2 Abgrenzung von System- und Prozessbausteine 16
2.3.3 Informationsverbund 17
2.3.4 Systembaustein APP.4.3 „Relationale Datenbanken“ 18
2.3.5 Systembaustein APP.6 Allgemeine Software 20
2.4 Kavalir (2021) 20
2.5 BSI-Bewertungsmaßstab in Kavalir (2021) 21
3. Analyse 23
3.1 Testsystem 23
3.2 Checkliste nach Kavalir (2021) 24
3.3 Beispiel für den Ablauf 26
3.3 Praktische Herausforderungen beim Bewerten 29
3.3 Auswertung 31
4. Fazit & Ausblick 35
4.1 Fazit Analyse 35
4.2 Fazit Methodologie 35
4.3 Ausblick 37
5. Quellenverzeichnis 39
6. Anhang 42

6.1 Checkliste nach Kavalir (2021) 42
6.2 Auswertung CouchDB.....45
6.3 Auswertung RethinkDB.....51
7. Selbstständigkeitserklärung.....56

Abbildungsverzeichnis

Abbildung 1 statista.com (2023) "Ranking of the most popular database management systems worldwide, as of September 2023" 7
Abbildung 2 Allianz Risk Barometer (2023) "Which cyber exposures concern your company most over the next year?" S.12 8
Abbildung 3 "Die möglichen drei Optionen des CAP-Theorems" aus Kaufmann (2023) S.163 9
Abbildung 4 "Beispiel einer Dokumentdatenbank" aus Kaufmann (2023) S.260 12
Abbildung 5 Logo "CouchDB" 12
Abbildung 6 Altes Logo RethinkDB 13
Abbildung 7 "IT-Grundschutz-Bausteine als Schichtenmodell" von BSI (2023) 17
Abbildung 8 "Lerneinheit 6.3." von BSI (2023) 22
Abbildung 9 "Einsatz der Bewertungskriterien" aus Kavalir (2021) S.82 22
Abbildung 10 Mustendatenbank ""maschinenbauprojekte" in MySQL 23
Abbildung 11 "Entity-Relationship-Modell" aus Kavalir (2021) S.111 24
Abbildung 12 Mustendatenbank ""maschinenbauprojekte" in CouchDB 24
Abbildung 13 Screenshot Konfigurationsdateien CouchDB 27
Abbildung 14 Screenshot Konfigurationsdatei RethinkDB 27
Abbildung 15 Replicator von CouchDB 27
Abbildung 16 Einrichten des Passwortes bei couchDB Installation 36
Abbildung 17 Python-Script zur Abfrage von Usern und Rechtevergabe in RethinkDB 38
Abbildung 18 Rollensetup in CouchDB 45
Abbildung 19 PGP Public Key von RethinkDB 55

Tabellenverzeichnis

Tabelle 1 "Aufbau des Testsystems" 23
Tabelle 2 Ergebnisse der Analyse 32
Tabelle 3 Ergebnisse von Kavalir (2021) 33
Tabelle 4 Zusammenfassung Ergebnis für Teilanalyse mit Daten aus Kavalir (2021) für MongoDB, OrientDB und Redis 34
Tabelle 5 Bewertungsmaßstäbe im Vergleich 37

1. Aufgabenstellung

Im Mittelpunkt dieser Arbeit steht die (Teil-)Analyse von nicht-relationalen Datenbankmanagementsystemen (DBMS) in Bezug auf die Implementierung von Sicherheitsfunktionen. Dies wird an den Beispielen von CouchDB und RethinkDB untersucht. Als Basis für diese Analyse soll das IT-Grundschutz-Kompendium des Bundesamts für Sicherheit in der Informationstechnik (BSI) dienen, welcher im Rahmen der Masterarbeit von Kavalir (2021) aufbereitet wurde. Da vom BSI weiterhin kein spezifischer Leitfaden für nicht-relationale Datenbankmanagement bereitgestellt wird, wurde der aufbereitete Systembaustein "APP.4.3: Relationale Datenbanken" als zentraler Bestandteil auch für diese Analyse herangezogen werden. Es soll auch überprüft werde, ob das Vorgehen sich übertragen lässt und welche Schwächen und Stärken wahrgenommen werden, so dass ggf. das Vorgehen verbessert werden kann. Die Analyse selbst deckt nicht den gesamten Informationsverbund ab, so dass diese Analyse im Vergleich zu Kavalir (2021) nur eine Teilanalyse sein kann.

2. Grundlagen

Die Datenverarbeitung hat sich im Laufe der Jahre rasant entwickelt. Insbesondere die Art und Weise, wie Daten gespeichert und abgerufen werden, hat sich grundlegend verändert. In diesem Abschnitt werden die Unterschiede zwischen relationalen und nicht-relationalen Datenbanken erläutert und ein Einblick in das CAP-Theorem sowie die ACID- und BASE-Eigenschaften gegeben. Darüber hinaus wird auf die Verbreitung von Nicht-Relationalen Datenbanken und auf die Rolle des Bundesamtes für Sicherheit in der Informationstechnik in Bezug auf Datenbanksicherheit eingegangen.
Laut Statista-Daten hatten Nicht-Relationale Datenbanken global einen Marktanteil von ca. 17% im September 2023. Rot markiert in der Grafik sind alle Nicht-Relationalen Datenbanken, auch NoSQL-Datenbanken, genannt. 83% sind andere Datenbankmanagementsysteme (DBMS).

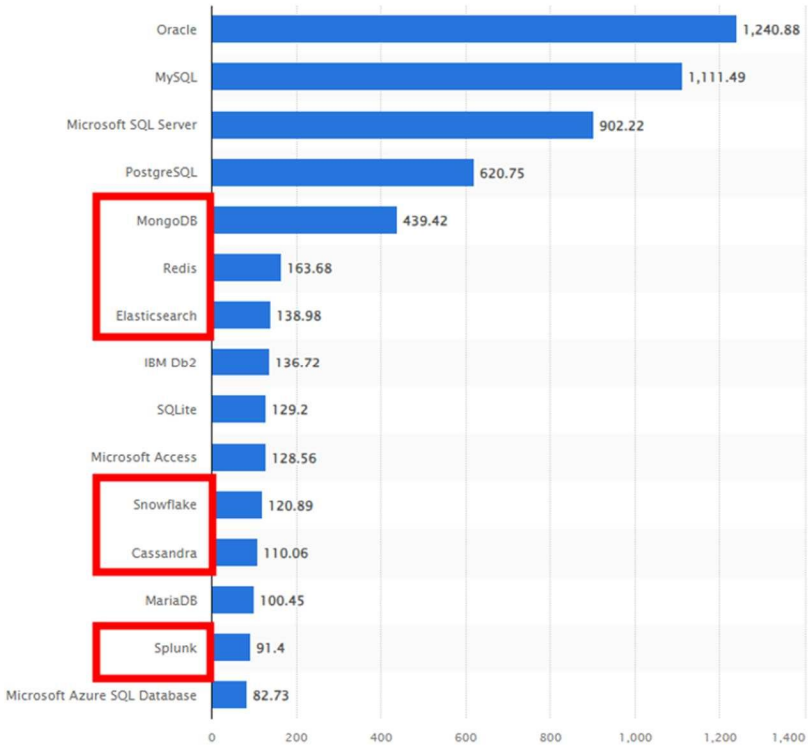

Abbildung 1 statista.com (2023) "Ranking of the most popular database management systems worldwide, as of September 2023"

Trotz der hohen Verbreitung deckt das Konzept des Bundamtes für Sicherheit in der Informationstechnik (BSI) weiterhin nicht NoSQL-Datenbanken ab. Gleichzeitig ist laut Allianz

Risk Barometer (2023)[1] weiterhin die Sorge auf Unternehmensseite vor Datenverlusten groß. Datenverluste sind entsprechend das größte wahrgenommene Sicherheitsrisiko für Unternehmen.

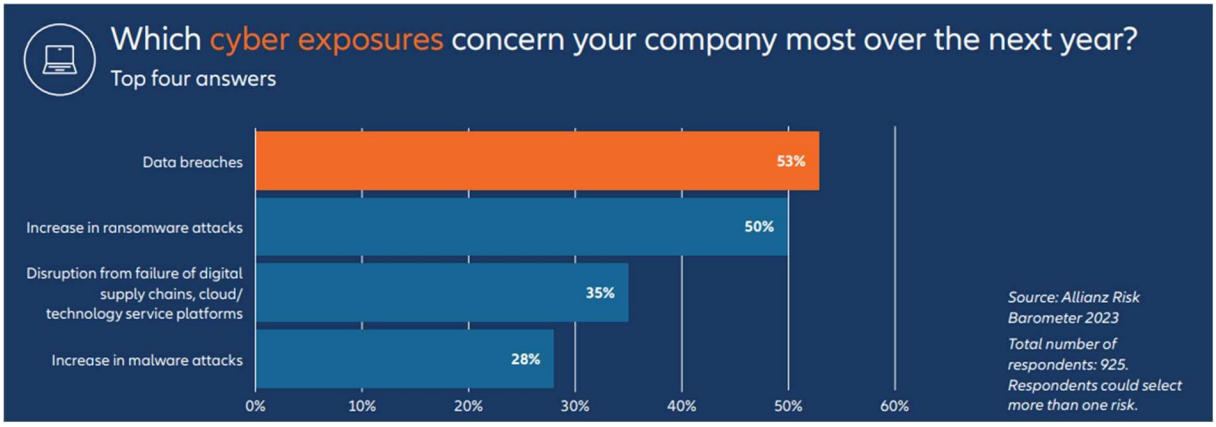

Abbildung 2 Allianz Risk Barometer (2023) "Which cyber exposures concern your company most over the next year?" S.12

Vor diesem Hintergrund ist die Frage nach der Sicherheit von Nicht-Relationalen Datenbanken weiterhin aktuell und relevant.

2.1 Abgrenzung Relationale Datenbanken von Nicht-Relationalen Datenbanken

Relationale und nicht-relationale Datenbanken haben ihre eigenen Vor- und Nachteile und eignen sich für unterschiedliche Anwendungsfälle. Die Wahl zwischen ihnen hängt von den spezifischen Anforderungen eines Projekts ab.
In der modernen Datenverarbeitung haben nicht-relationale Datenbanksysteme, häufig als NoSQL-Datenbanken bezeichnet[2], an Bedeutung gewonnen. Sie bieten Lösungen für Anwendungsfälle, die sich durch große Datenmengen, hohe Schreib- und Leseanforderungen oder die Notwendigkeit einer flexiblen Schemastruktur auszeichnen. Im Gegensatz zu den fest strukturierten relationalen Datenbanksystemen sind NoSQL-Datenbanken durch ihre Vielseitigkeit und Skalierbarkeit gekennzeichnet. Es gibt vier Haupttypen von NoSQL-Datenbanken: Dokumentenbasierte, Schlüssel-Wert, Spaltenorientierte und Graphdatenbanken. Jeder Typ hat seine spezifischen Eigenschaften und eignet sich für

[1] Allianz Risk Barometer (2023) S.6ff
[2] Kaufmann (2023) S. 13f

unterschiedliche Anwendungen. NoSQL-Datenbanken bieten eine flexible und skalierbare Alternative zu relationalen Datenbanksystemen.

2.1.1 CAP-Theorem

Das CAP-Theorem von Eric Brewer besagt, dass es in verteilten Datenbank-Systemen nicht möglich ist, gleichzeitig Konsistenz (Consistency), Verfügbarkeit (Availability) und Partitionstoleranz (Partition Tolerance) zu gewährleisten.[3]

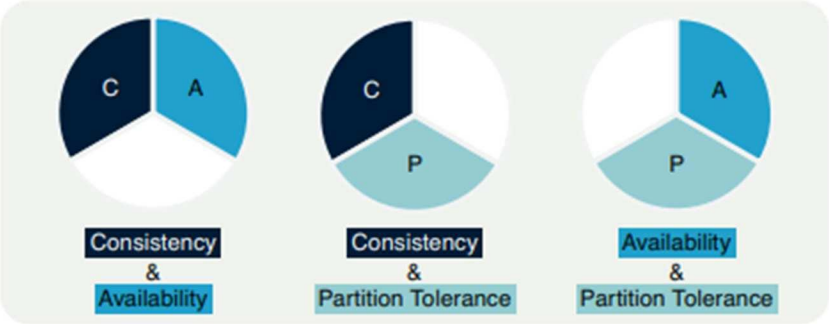

Abbildung 3 "Die möglichen drei Optionen des CAP-Theorems" aus Kaufmann (2023) S.163

Ein System kann immer nur zwei dieser drei Eigenschaften erfüllen.

- Konsistenz: Alle Knoten sehen zur gleichen Zeit die gleichen Daten.
- Verfügbarkeit garantiert, dass jede Anfrage entweder ein Ergebnis zurückgibt oder einen Fehler meldet.
- Partitionstoleranz: Das System funktioniert weiterhin, auch wenn die Kommunikation zwischen den Knoten gestört ist.

Das Begriffspaar „ACID“ und „BASE“ beschreibt die Gegensätzlichkeit von Relationalen und Nicht-Relationalen Datenbanken anhand ihrer Eigenschaften in Bezug auf ihre Datenorganisation und ihren Umgang mit den Daten selbst.
Zusammenfassend lässt sich sagen, dass ACID und BASE unterschiedliche Philosophien und Ansätze darstellen, wenn es um das Management von Datenbanktransaktionen geht.

[3] Schicker (2017) S.312f

Während ACID strikte Regeln und Garantien bietet, bietet BASE mehr Flexibilität, kann aber zu vorübergehenden Inkonsistenzen führen. Die Wahl zwischen den beiden hängt von den spezifischen Anforderungen und dem Kontext des Systems ab.

2.1.2 ACID

ACID ist ein Akronym, das für Atomicity, Consistency, Isolation und Durability steht und in der Regel mit relationalen Datenbanksystemen in Verbindung gebracht wird.[4][5]

- Atomicity (Atomarität) bezieht sich auf das "Alles-oder-Nichts"-Prinzip. Das bedeutet, dass eine Transaktion, die aus mehreren Operationen besteht, entweder vollständig ausgeführt wird oder, wenn irgendein Teil der Transaktion fehlschlägt, die gesamte Transaktion rückgängig gemacht wird. Dies stellt sicher, dass die Datenbank zu jeder Zeit in einem gültigen Zustand bleibt.
- Consistency (Konsistenz) gewährleistet, dass jede erfolgreich abgeschlossene Transaktion die Datenbank von einem konsistenten Zustand in einen anderen konsistenten Zustand überführt. Selbst wenn während der Transaktion Inkonsistenzen auftreten, muss am Ende der Transaktion Konsistenz gewährleistet sein.
- Mit Isolation (Isolierung) wird sichergestellt, dass gleichzeitig laufende Transaktionen sich nicht gegenseitig beeinflussen. Dies bedeutet, dass die Ergebnisse einer Transaktion für andere Transaktionen erst sichtbar werden, wenn sie abgeschlossen ist.
- Durability (Dauerhaftigkeit) garantiert, dass einmal durchgeführte Transaktionen dauerhaft in der Datenbank gespeichert sind, selbst im Falle von Systemausfällen. Dies wird oft durch den Einsatz von Transaktionslogs erreicht, mit denen sich Änderungen wiederherstellen lassen.

[4] Kaufmann (2023) S.150f
[5] Schicker (2017) S. 18f

2.1.3 BASE

Dem gegenüber steht BASE, das eher mit nicht-relationalen Datenbanken assoziiert wird und für Basically Available, Soft state und Eventually consistent steht.[6][7]

- Basically Available (Grundsätzlich verfügbar) betont die Verfügbarkeit des Systems. Während es Zeiten geben kann, in denen das System aufgrund von Netzwerkpartitionen oder anderen Fehlern kurzzeitig inkonsistent ist, bleibt es dennoch zugänglich und betriebsbereit.
- Soft state (Weicher Zustand) bedeutet, dass der Zustand des Systems sich über die Zeit hinweg ändern kann, selbst wenn keine neuen Transaktionen stattfinden. Das System kann also fluktuieren, je nachdem, wie es sich anpasst und wie die Daten repliziert werden.
- Eventually consistent (Letztlich konsistent) ist vielleicht das Schlüsselprinzip von BASE. Es besagt, dass das System nach einer gewissen Zeit und nachdem alle Transaktionen verarbeitet wurden, einen konsistenten Zustand erreicht. Diese Konsistenz wird jedoch nicht sofort oder nach einer festgelegten Zeitspanne garantiert, sondern "letztendlich".

2.2 Arten von NoSQL-Datenbanken

2.2.1 Dokumentenbasierte Datenbanken

Dokumentenbasierte Datenbanken speichern ihre Daten in dokumentartigen Strukturen, oft im JSON- oder BSON-Format. Diese Dokumente können unterschiedliche Felder und Datenstrukturen aufweisen, was eine hohe Flexibilität in Bezug auf das Datenbankschema ermöglicht.

[6] Kaufmann (2023) S. 162f
[7] Schicker (2017) S. 314f

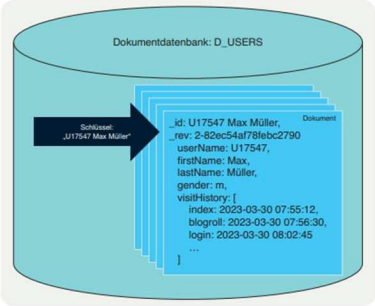

Abbildung 4 "Beispiel einer Dokumentdatenbank" aus Kaufmann (2023) S.260

Ein prominentes Beispiel für eine dokumentenbasierte Datenbank ist MongoDB.[8] Diese Art von Datenbank eignet sich besonders für Anwendungsfälle, bei denen die Datenstruktur variieren kann und ein festes Schema nicht notwendig oder sogar hinderlich ist.[9]

2.2.1.1 CouchDB

CouchDB ist eine dokumentenbasierte NoSQL-Datenbank, die unter der Schirmherrschaft der Apache Software Foundation steht. Sie wurde im Februar 2005 veröffentlicht.

Abbildung 5 Logo "CouchDB"[10]

CouchDB zeichnet sich Zuverlässigkeit und Skalierbarkeit aus. Der Hauptvorteil von CouchDB ist die Fähigkeit, Daten in JSON-Format zu speichern, was Entwicklern eine erhebliche Flexibilität in Bezug auf die Datenmodellierung bietet. Mit der HTTP-basierten API ermöglicht CouchDB eine einfache Integration in Web-Anwendungen. Darüber hinaus ist die Multi-Master-Replikation ein wichtiges Merkmal, das die Datenreplikation über verschiedene Instanzen und Geräte hinweg erleichtert. Im Laufe der Jahre hat CouchDB bedeutende

[8] Schicker (2017) S. 315
[9] Kaufmann (2023) S. 21ff
[10] https://de.wikipedia.org/wiki/Datei:Apache_CouchDB_logo.svg (letzter Aufruf 26.10.2023)

Entwicklungen durchgemacht, darunter 2010 die Einführung der Version 1.0 und 2016 die Veröffentlichung von CouchDB 2.0 mit einer neuen, clusterfähigen Speicher-Engine.[11]

2.2.1.2 RethinkDB

RethinkDB hebt sich von anderen Datenbanken durch seine Echtzeitfunktionen ab. RethinkDB wurde im November 2012 veröffentlicht.

Abbildung 6 Altes Logo RethinkDB

Die Echtzeit-Push-Architektur unterscheidet RethinkDB von vielen traditionellen Datenbanken, die meistens eine "Pull"-Methode zur Datenabfrage verwenden. RethinkDB speichert Daten ebenfalls als JSON-Dokumente, was eine flexible Datenmodellierung ermöglicht. Diese Flexibilität wird durch die Verteilungsarchitektur von RethinkDB weiter verstärkt, die einen Fokus auf horizontale Skalierbarkeit legt und eine Verwaltung über ein Web-Interface bietet. Die Firma hinter RethinkDB musste im Oktober 2016 Insolvenz anmelden. [12] Heute wird RethinkDB von einer Community[13] aufrechterhalten und weiterentwickelt.[14] Der Scoure-Code ist seit Juli 2017 wieder openscource.

2.2.2 Schlüssel-Wert-Datenbanken

Schlüssel-Wert-Datenbanken nutzen eine einfache Datenmodellierung, bei der jedem Schlüssel ein spezifischer Wert zugeordnet wird. Dieser Wert kann eine Zeichenfolge, eine Zahl oder auch ein komplexeres Objekt sein. Da der Zugriff auf den Wert direkt über den Schlüssel erfolgt, sind diese Datenbanken besonders schnell in Lese- und Schreiboperationen.

[11]Seite: https://de.wikipedia.org/wiki/CouchDB (letzter Aufruf 26.10.2023)
[12] Seite: https://dbdb.io/db/rethinkdb (letzter Aufruf 26.10.2023)
[13] Seite: https://github.com/rethinkdb/rethinkdb (letzter Aufruf 26.10.2023)
[14] Seite: https://de.wikipedia.org/wiki/RethinkDB (letzter Aufruf 26.10.2023)

Beispiele für Schlüssel-Wert-Datenbanken sind Redis und Riak. Sie eignen sich insbesondere für Anwendungen mit hohem Durchsatz und einfacher Datenstruktur.[15][16]

2.2.3 Spaltenorientierte Datenbanken

Spaltenorientierte Datenbanken, auch als Wide-Column-Stores bekannt, speichern ihre Daten in Spalten und nicht in Zeilen, wie es bei traditionellen relationalen Datenbanken der Fall ist. Dies ermöglicht eine effizientere Abfrage und Speicherung großer Datenmengen. Cassandra und HBase sind prominente Vertreter dieser Kategorie.[17] Sie sind besonders geeignet für Anwendungen, die große Mengen von schnell veränderlichen Daten verarbeiten, wie beispielsweise Time-Series-Daten.[18]

2.2.4 Graphdatenbanken

Graphdatenbanken legen ihren Fokus auf die Beziehungen zwischen den Daten. Sie speichern Entitäten als Knoten und die Beziehungen zwischen ihnen als Kanten. Damit ermöglichen sie effiziente Abfragen von komplex vernetzten Datenstrukturen. Neo4j ist ein bekanntes Beispiel für eine Graphdatenbank. Diese Art von Datenbank eignet sich besonders für Anwendungen, die komplexe Beziehungsstrukturen abbilden, wie soziale Netzwerke oder Empfehlungssysteme.[19][20]

2.3 Bundesamt für Sicherheit in der Informationstechnik (BSI)

Das BSI ist eine zentrale Institution in Deutschland, die sich mit der Sicherheit in der Informationstechnik befasst.
Hierbei fungiert das BSI als Regulierungs- und Überwachungsbehörde, indem es Standards und Richtlinien für die Sicherheit in der Informationstechnik festlegt und die Einhaltung dieser Normen durch verschiedene Organisationen und Unternehmen überwacht. Darüber

[15] Kaufmann (2023) S. 253ff
[16] Schicker (2017) S. 315
[17] Schicker (2017) S. 315f
[18] Kaufmann (2023) S.256ff
[19] Kaufmann (2023) S.16ff
[20] Schicker (2017) S. 316

hinaus bietet es Beratung und Unterstützung für Regierungsbehörden, Unternehmen sowie die breite Öffentlichkeit in Fragen der IT-Sicherheit an. In dem Bestreben, die IT-Sicherheit kontinuierlich zu verbessern, beteiligt sich das BSI aktiv an Forschung und Entwicklung, um neue Sicherheitstechnologien zu fördern und aufkommende Bedrohungen zu identifizieren. Bei großen IT-Sicherheitsvorfällen tritt das BSI in Aktion und bietet Krisenmanagement-Unterstützung, koordiniert die Reaktionen auf diese Vorfälle und hilft bei der Bewältigung der Situation. Um das Bewusstsein für IT-Sicherheitsrisiken zu erhöhen, führt es Aufklärungs- und Bildungsmaßnahmen durch und fördert Best Practices in diesem Bereich. Auf internationaler Ebene arbeitet das BSI mit Partnern und Organisationen zusammen, um globale IT-Sicherheitsstandards zu fördern und grenzüberschreitende Bedrohungen effektiv zu bekämpfen. Im Rahmen seiner Aufgaben bietet das BSI auch Zertifizierungsdienste für IT-Produkte und -Systeme an, um deren Sicherheit und Konformität mit nationalen und internationalen Standards zu gewährleisten. Durch verschiedene Initiativen und Programme engagiert sich das BSI aktiv für die Förderung der IT-Sicherheit in Deutschland und trägt maßgeblich zur Entwicklung einer sicheren Informationsgesellschaft bei.

2.3.1 IT-Grundschutz-Kompendium des BSI

Das IT-Grundschutz-Kompendium[21] stellt einen Leitfaden dar, der von dem Bundesamt für Sicherheit in der Informationstechnik (BSI) herausgegeben wird, um Organisationen bei der Identifizierung und Umsetzung von Sicherheitsmaßnahmen zu unterstützen. Die Hauptzielsetzung des IT-Grundschutz-Kompendiums liegt in der Bereitstellung eines methodischen Ansatzes zur Erhöhung der Informationssicherheit.[22]
Ein zentraler Aspekt des Kompendiums ist die Standardisierung, da es standardisierte Vorgehensweisen und Maßnahmen bereitstellt, um eine einheitliche Umsetzung der Informationssicherheit über verschiedene Organisationen hinweg zu fördern. Weiterhin legt das Kompendium großen Wert auf ein systematisches Risikomanagement, um potenzielle Sicherheitsrisiken zu identifizieren und geeignete Maßnahmen zur Minderung dieser Risiken zu ergreifen. Dabei steht die praktische Umsetzbarkeit der empfohlenen Maßnahmen im Vordergrund, um eine realistische und effektive Verbesserung der Sicherheitslage zu

[21] BSI (2023) „IT-Grundschutz Kompendium“
[22] BSI (2023) „IT-Grundschutz-Kompendium – Werkzeug für Informationssicherheit Edition 2023“

ermöglichen. Durch die Bereitstellung von Information und Anleitung fördert das Kompendium das Bewusstsein für Informationssicherheit und die Bedeutung geeigneter Schutzmaßnahmen. Sein modularer Aufbau ermöglicht eine flexible Anpassung an die spezifischen Anforderungen und Gegebenheiten verschiedener Organisationen, wodurch eine maßgeschneiderte Umsetzung der Sicherheitsmaßnahmen ermöglicht wird. Das IT-Grundschutz-Kompendium wird regelmäßig aktualisiert, um auf neue Bedrohungen und Entwicklungen im Bereich der Informationssicherheit zu reagieren, und stellt somit eine aktuelle und anpassungsfähige Ressource dar. Zudem dient das Kompendium als Grundlage für die Zertifizierung nach IT-Grundschutz, wodurch es einen anerkannten Standard für Informationssicherheit aufzeigt und somit maßgeblich dazu beiträgt, ein hohes Niveau an Informationssicherheit in Organisationen zu fördern und zu erhalten.

2.3.2 Abgrenzung von System- und Prozessbausteine

Im Kontext des IT-Grundschutzes werden System- und Prozessbausteine als Modellierungselemente verwendet, um die Komplexität der Informationsverarbeitung in Organisationen strukturiert darzustellen und zu analysieren. Systemsteine repräsentieren die technischen Komponenten wie Hardware, Software und Netzwerke, die in einem Informationssystem integriert sind. Sie helfen dabei, eine klare Übersicht über die technische Architektur und die verwendeten Technologien zu erhalten, und sind somit entscheidend für die Identifikation von Sicherheitsanforderungen und -maßnahmen.

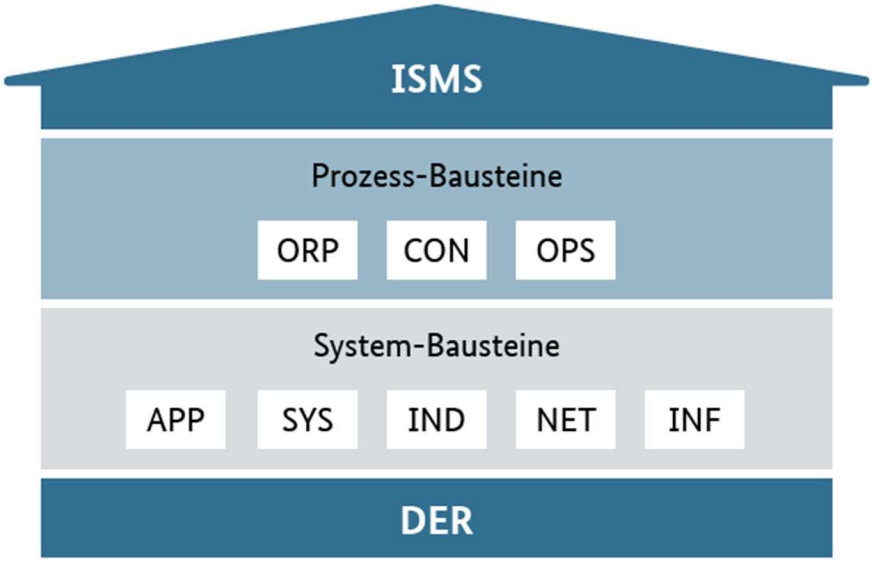

Abbildung 7 "IT-Grundschutz-Bausteine als Schichtenmodell" von BSI (2023)[23]

Auf der anderen Seite stehen die Prozesssteine, die die organisatorischen und geschäftsprozessbezogenen Aspekte der Informationsverarbeitung repräsentieren. Sie helfen dabei, die Art und Weise zu verstehen, wie Informationen in einer Organisation fließen, verarbeitet und genutzt werden. Prozesssteine erfassen auch die Rollen und Verantwortlichkeiten der beteiligten Akteure, und sind somit unerlässlich für die Gestaltung und Optimierung von Geschäftsprozessen im Hinblick auf die Informationssicherheit. Zusammen bilden System- und Prozesssteine ein holistisches Modell, das eine umfassende Sicht auf die Informationsverarbeitung in einer Organisation ermöglicht. Durch die Verwendung von System- und Prozesssteinen können Organisationen ihre Informationsinfrastruktur und Geschäftsprozesse besser verstehen und entsprechende Maßnahmen zur Verbesserung der Informationssicherheit ergreifen.

2.3.3 Informationsverbund

Der Begriff "Informationsverbund"[24] im Kontext des BSIs bezieht sich auf die strukturierte Zusammenfassung von Komponenten und Prozessen, die gemeinsam zur Informationsverarbeitung und -kommunikation beitragen. Ein Informationsverbund kann aus verschiedenen Elementen bestehen, darunter Netzwerkinfrastrukturen,

[23] BSI (2023) „Lerneinheit 5.2: Schichtenmodell“
[24] BSI (2023) „Lerneinheit 2.8: Das Sicherheitskonzept“

Informationsverarbeitungssysteme, Anwendungen sowie organisatorische Strukturen und Prozesse. Ziel des Informationsverbunds ist es, eine organisierte und kontrollierte Umgebung zu schaffen, in der Informationen sicher verarbeitet, gespeichert und übertragen werden können. Im Rahmen des IT-Grundschutzes wird der Informationsverbund als eine Einheit betrachtet, für die ein einheitliches Sicherheitskonzept erarbeitet wird, um die Informationssicherheit auf einem akzeptablen Niveau zu halten. Dabei werden die Sicherheitsanforderungen und -maßnahmen für den gesamten Verbund festgelegt, um eine kohärente und effektive Sicherheitsarchitektur zu gewährleisten. Ein gut definierter und verwalteter Informationsverbund ist entscheidend für die Umsetzung und Aufrechterhaltung der Informationssicherheit innerhalb einer Organisation, indem er hilft, Risiken zu identifizieren, zu bewerten und geeignete Schutzmaßnahmen zu implementieren. Im Gegensatz zu Kavalir (2021), wo für die Analyse ein Informationsverbund modelliert wurde, wurde im Rahmen dieser Arbeit auf die Modellierung eines Informationsverbundes verzichtet. Dadurch können nur Systembausteine und nicht Prozesssteine berücksichtigt (vgl. Abbildung 7). Für diese Teilanalyse wurde entsprechend nur die Systembausteine APP.4.3 und ein Unterpunkt aus APP.6 berücksichtigt.

2.3.4 Systembaustein APP.4.3 „Relationale Datenbanken“

Die wesentlichen Inhalte für die Analyse basieren auf diesen Baustein. Der Baustein APP.4.3 legt spezifische Anforderungen fest, um Gefährdungen zu reduzieren und die Sicherheit von Relationalen Datenbanksystemen zu verbessern.[25] Die typische Gefährdungslage umfasst verschiedene Szenarien, beispielsweise unzureichende Dimensionierung der Systemressourcen, die zu Ausfällen oder fehlerhafter Funktion der Datenbank führen kann. Aktivierte Standard-Konten und unverschlüsselte Datenbankanbindungen können die Sicherheit kompromittieren, indem sie unbefugten Zugriff oder Datenmanipulation ermöglichen. Datenverlust, Integritätsverlust der gespeicherten Daten, SQL-Injections, unsichere Konfiguration des DBMS und Malware bzw. unsichere Datenbank-Skripte stellen weitere potenzielle Bedrohungen dar, die die Vertraulichkeit, Integrität und Verfügbarkeit der Daten gefährden können.

[25] BSI (2023) „APP.4.3 Relationale Datenbanken“

Folgende Elemente des Bausteins APP.4.3 sind Teil der Checkliste nach Kavalir (2021)[26] und sind ein Teil der Analyse.

- A3 Basishärtung (Basis)
- A9 Datensicherung (Basis)
- A13 Handhabung DB-Links (Standard)
- A16 Verschlüsselung Verbindungen (Standard)
- A18 Überwachung (Standard)
- A19 Schutz vor Datenbankskripten (Standard)
- A21 Einsatz von Security Tools (Erhöhter Schutzbedarf)
- A24 Data-at-Rest Verschlüsselung (Erhöhter Schutzbedarf)

Aus den spezifischen Inhalten der Teilbausteine wurde eine Fragestellung für die Analyse mittels Checkliste entwickelt. Diesmal soll einmal beispielhaft dargelegt werden.

> *„APP.4.3.A9 Datensicherung eines Datenbanksystems (B)*
> *Es MÜSSEN regelmäßig Systemsicherungen des DBMS und der Daten durchgeführt werden. Auchbevor eine Datenbank neu erzeugt wird, MUSS das Datenbanksystem gesichert werden. Hierfür SOLLTEN die dafür zulässigen Dienstprogramme benutzt werden. Alle Transaktionen SOLLTEN so gesichert werden, dass sie jederzeit wiederherstellbar sind. Wenn die Datensicherung die verfügbaren Kapazitäten übersteigt, SOLLTE ein erweitertes Konzept erstellt“* aus BSI (2023) „APP.4.3 Relationale Datenbanken“

In Kavalir (2021) wurden hieraus folgende Fragen abgeleitet:[27]

- Welche Funktionen zur Systemsicherung von DBMS und Daten sind vorhanden?
- Wird das Einrichten einer automatischen Systemsicherung unterstützt?
- Sind die Transaktionen wiederherstellbar?
- Gibt es eine Funktion zur inkrementellen Sicherung?

[26] Kavalir (2021) S.33ff
[27] Kavalir (2021) S.38f

- Welche Parameter können vorgegeben werden?

Diese Fragen wurden in eine Checkliste überführt und im Rahmen der Analyse überprüft. Da sich die Inhalte des Bausteins auf Relationale Datenbanken beziehen, tauchen auch Inhalte auf, welche in keinen direkten Zusammenhang zu Nicht-Relationalen Datenbanken bestehen, beispielsweise die Frage „Sind die Transaktionen wiederherstellbar?“. Diese Frage kann ich Bezug auf Nicht-Relationale Datenbanken aufgrund des BASE-Prinzips im Regelfall verneint werden (siehe auch 2.1.1 CAP-Theorem S. 9).

2.3.5 Systembaustein APP.6 Allgemeine Software

Ziel des Bausteins ist es, Sicherheitsanforderungen zu identifizieren und umzusetzen, um die Software und die damit verarbeiteten Informationen über den gesamten Lebenszyklus hinweg zu schützen.[28] Er konzentriert sich auf standardisierte und generische Verfahren im Software-Lebenszyklus, ohne konkrete Konfigurations- oder Schutzempfehlungen zu geben. Einige spezifische Aspekte des Software-Lebenszyklus werden in anderen Bausteinen behandelt, wie z.B. Software-Tests und Patch-Management.
Die dargestellte Gefährdungslage umfasst verschiedene Szenarien, darunter ungeeignete Software-Auswahl, fehlerhafte Konfiguration, Bezug von Software aus unzuverlässigen Quellen, mangelhafte Wartung, fehlerhafte Nutzung, unzureichende Ressourcen für die Software-Ausführung und Nichtbeachtung der Anforderungen der Benutzer. Für die Analyse von Relevanz ist der Unterpunkt „A3 Beschaffung und Integrität (Basis)“.

2.4 Kavalir (2021)

In der Masterarbeit von Sebastian Kavalir, eingereicht an der Hochschule Wismar im Oktober 2021, wurde eine sicherheitstechnische Untersuchung von NoSQL-Datenbankmanagementsystemen (DBMS) durchgeführt.[29] Die Untersuchung konzentrierte sich auf drei spezifische DBMS: OrientDB, MongoDB und Redis, mit dem Ziel, eine breite Variation innerhalb der NoSQL-DBMS zu erforschen. Als Basis für die Untersuchung der

[28] BSI (2023) „APP.6 Allgemeine Software“
[29] Kavalir (2021)

Härtungsmaßnahmen diente das IT-Grundschutz-Kompendium des Bundesamtes für Sicherheit in der Informationstechnik (BSI), insbesondere der Systembaustein "APP.4.3 - Relationale Datenbankmanagementsysteme", der als Checkliste für die Härtung von DBMS angesehen werden kann. Durch die Anwendung des IT-Grundschutz-Kompendiums zur Konzeption eines Managementsystems für Informationssicherheit (ISMS) wurde ein Auszug eines Informationsverbundes beispielhaft erarbeitet. Bei technisch realisierbaren Anforderungen wurde die spezifische Umsetzung an den betrachteten DBMS dargestellt, während für organisatorische Anforderungen Umsetzungsvorschläge entwickelt wurden. Ausgehend von der Checkliste, welche in Kavalir (2021) detailliert erarbeitet wurde, stellte sich die Frage, ob das Vorgehen aus der Masterarbeit auf weitere DBMS übertragbar ist. Da aber im Rahmen dieser Projektarbeit kein Informationsverbund abgebildet werden kann, ist der Fokus auf einen Teilaspekt von Kavalirs Analyse ausgerichtet. Hierbei handelt es sich um den Teil, welcher die Software und ihre Qualität in Bezug auf Sicherheit näher betrachtet hat. Die vorgestellten Systembausteine APP.4.3 und APP.6 bilden den Kern dieser Checkliste. Die Checkliste aus Kavalir (2021) wird im Anhang (6.1 Checkliste nach Kavalir (2021) S. 42) vollständig dokumentiert.

2.5 BSI-Bewertungsmaßstab in Kavalir (2021)

Bei der Bewertung des Umsetzungsgrades von IT-Grundschutz-Anforderungen in Bezug auf bestimmte Zielobjekte werden unterschiedliche Kategorien herangezogen. Eine Anforderung kann als "entbehrlich" eingestuft werden, wenn ihre Umsetzung nicht zwingend notwendig ist. Dies kann etwa der Fall sein, wenn alternative Schutzmaßnahmen vorhanden sind, die ebenso effektiv gegen mögliche Gefährdungen vorgehen, oder wenn die Anforderung für den konkreten Anwendungsfall nicht relevant ist. Wenn eine Anforderung durch entsprechende Maßnahmen in vollem Maße erfüllt wird, wird sie mit "ja" gekennzeichnet. Ist die Umsetzung nur teilweise erfolgt, wird die Kategorie "teilweise" verwendet. Schließlich wird die Anforderung mit "nein" bewertet, wenn sie nicht umgesetzt wurde und die notwendigen Maßnahmen zum Großteil fehlen.

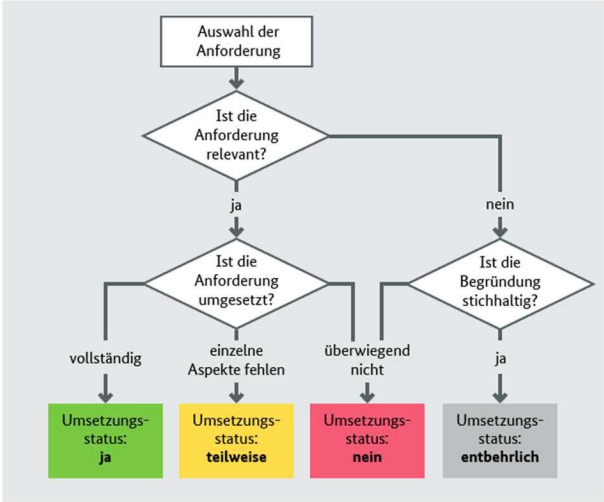

Abbildung 8 "Lerneinheit 6.3." von BSI (2023)

Als Basis für die Bewertungskriterien diente die Checkliste des BSI zum Baustein APP4.3, der sich auf relationale Datenbanksysteme bezieht. Das BSI unterscheidet vier Kategorien: "entbehrlich" für nicht notwendige Umsetzungen, "ja" für vollständige Erfüllung der Anforderungen, "teilw." für teilweise Erfüllung und "nein" für fehlende Umsetzung. Für die Auswertung nach Kavalir (2021)[30] wurde allerdings das Kriterium "entbehrlich" nicht berücksichtigt (siehe Abbildung 9).

Kategorie	Beschreibung	Zahlenwert
„ja“	Vollständige Umsetzung im DBMS möglich	2
„teilw.“	Teilweise Umsetzung im DBMS möglich	1
„nein“	Keine Umsetzung im DBMS möglich	0

Abbildung 9 "Einsatz der Bewertungskriterien" aus Kavalir (2021) S.82

Die restlichen Kategorien wurden mit Zahlenwerten versehen: 2 Punkte für "ja", 1 Punkt für "teilw." und 0 Punkte für "nein". Es gab zudem eine Unterscheidung der Anforderungen in technische und organisatorische. Für die Bewertung im Rahmen dieser Projektarbeit wurden ausschließlich technische Anforderungen berücksichtigt, da kein Informationsverbund vorliegt und eine Bewertung der organisatorischen Anforderungserfüllung durch das DBMS nicht möglich ist.

[30] Kavalir (2021) S. 82

3. Analyse

3.1 Testsystem

Für diese Arbeit wurden Testsysteme aufgebaut, auf diesen dann die zu überprüfende Software installiert wurde. Während in Kavalir (2021) ein Informationsverbund nachgebaut wurde (siehe 2.3.3 Informationsverbund S. 17), wurde für diese Teilanalyse nur die Software auf einem Testsystem installiert.

Windows 11 (Host)	
Virtuelle Maschine 1 (VM1)	Virtuelle Maschine 2 (VM2)
Ubuntu 22.04.3	Ubuntu 22.04.3
CouchDB v.3.3.2	RethinkDB v.2.4.3-0jammy

Tabelle 1 "Aufbau des Testsystems"

RethinkDB wurde darüber hinaus mit Python installiert.[31][32] Um Funktionen wie die Datensicherung zu testen, wurde erneut analog zu Kavalir (2021). Es wurde in beiden DBMS Musterdatenbanken erstellt.

Abbildung 10 Mustendatenbank ""maschinenbauprojekte" in MySQL

Hierzu wurde zuerst in MySQL das Entity-Relationship Modell nachgebildet und dann in die verschiedenen Datenbanken exportiert.

[31] https://rethinkdb.com/docs/install-drivers (letzter Aufruf 26.10.2023)
[32] https://rethinkdb.com/docs/install-drivers/python/ (letzter Aufruf 26.10.2023)

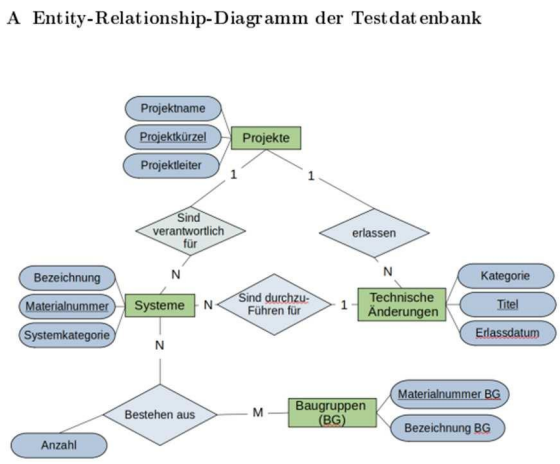

Abbildung 11 "Entity-Relationship-Modell" aus Kavalir (2021) S.111

Beim Export wurde auf die ursprüngliche Tabellenstruktur verzichtet.

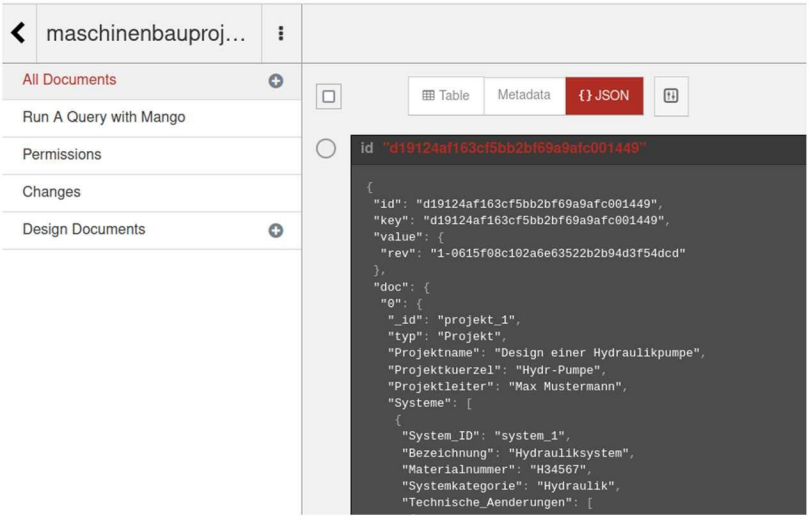

Abbildung 12 Mustendatenbank ""maschinenbauprojekte" in CouchDB

3.2 Checkliste nach Kavalir (2021)

Die aufgelisteten Kriterien aus Kavalir (2021), welche im Anhang dokumentiert sind (siehe 6.1 Checkliste nach Kavalir (2021) S. 42), bieten einen detaillierten Einblick in die Sicherheitsaspekte von nicht-relationalen Datenbankmanagementsystemen (DBMS). Beispielhaft wird dies mittels CouchDB und RethinkDB überprüft, in Übereinstimmung mit dem IT-Grundschutz-Kompendium des Bundesamts für Sicherheit in der Informationstechnik (BSI). Initial konzentrieren sich die Maßnahmen auf die Eliminierung unnötiger Default-Einstellungen und Funktionen in den Datenbanksystemen, um die Angriffsfläche zu

minimieren. Es folgt eine konsequente Restriktion von Datenbankfunktionen, die eine potenzielle Interaktion mit dem Betriebssystem oder anderen Netzwerkdiensten ermöglichen, und somit die Systemsicherheit gefährden könnten.

Eine weitere wesentliche Maßnahme ist die sorgfältige Konfiguration von Berechtigungen und Benutzerkonten im Datenbanksystem, inklusive der Implementierung robuster Authentifizierungsmaßnahmen. Insbesondere die Anforderungen an Passwörter und Authentifizierungsmerkmale werden dabei hervorgehoben, um einem hohen Sicherheitsstandard Rechnung zu tragen.

In den folgenden Punkten wird die Bedeutung von allgemeinen Best Practices für die Datenbankverwaltung und -konfiguration betont, mit einem speziellen Fokus auf der korrekten Konfiguration der Logging-Funktionalitäten und der systematischen Überwachung von Datenbanksystemen.

Die Netzwerk- und Kommunikationssicherheit wird ebenfalls adressiert, einschließlich der korrekten Konfiguration von SSL und anderen relevanten Verbindungseinstellungen. Hinsichtlich der Datensicherung und Wiederherstellung werden spezielle Funktionen und Parameter für die Systemsicherung von DBMS und Daten aufgeführt.

Ein umfangreicher Abschnitt widmet sich verschiedenen Aspekten der Datenbanksicherheit, einschließlich der Verschlüsselung, der Überwachung des Datenbankmanagementsystems und der Handhabung potenziell schädlicher Datenbank-Skripte. Besondere Aufmerksamkeit wird der Verschlüsselung gewidmet, wobei unterschiedliche Verschlüsselungsansätze und ihre jeweiligen Auswirkungen auf die Performance, Schlüsselverwaltungsprozesse und Backup-Recovery-Konzepte diskutiert werden.

Abschließend wird die sichere Beschaffung von Software und die Verschlüsselung der Datenbankanbindung hervorgehoben, um ein umfassendes Verständnis der erforderlichen Maßnahmen zur Gewährleistung der Sicherheit in nicht-relationalen DBMS zu vermitteln.

3.3 Beispiel für den Ablauf

Ausgangspunkt für die Analyse ist die Detail-Anforderung des Systembausteins. Aus dieser wurde eine konkrete Fragestellung abgeleitet.

> *„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. 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. Abhängig vom Schutzbedarf der Daten SOLLTEN die Wiederherstellungsparameter vorgegeben werden.“*
> APP.4.3.A9 – „Datensicherung eines Datenbanksystems“

Die konkreten Fragen, welche aus dem Inhalt des Bausteins abgeleitet wurden, stellen dann den Ausgangspunkt für die tatsächliche Analyse dar.

- Welche Funktionen zur Systemsicherung von DBMS und Daten sind vorhanden?
- Wird das Einrichten einer automatischen Systemsicherung unterstützt?
- Sind die Transaktionen wiederherstellbar?
- Gibt es eine Funktion zur inkrementellen Sicherung?
- Welche Parameter können vorgegeben werden?

Mit Hilfe der Fragen wird dann die Software untersucht. Hierzu werden dann neben dem Frontend-Einstellungen selbstverständlich auch die Konfigurationsdateien überprüft. Dazu wird auch die Software-Dokumentation überprüft.

Für CouchDB wurde direkt nach der Installation Backups von „default.ini“ und „local.ini“ angefertigt.

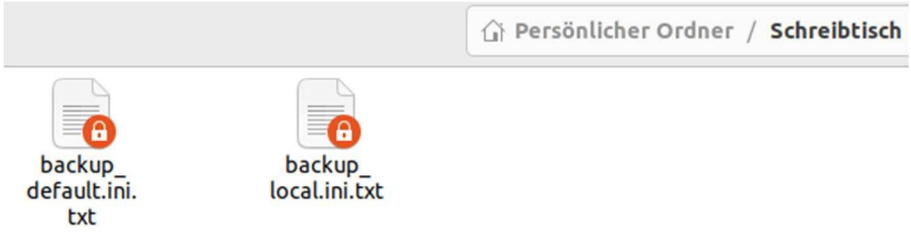

Abbildung 13 Screenshot Konfigurationsdateien CouchDB

Bei RethinkDB ist es „instance1.conf“, die nach der Installation mittels Backups gesichert wurde.

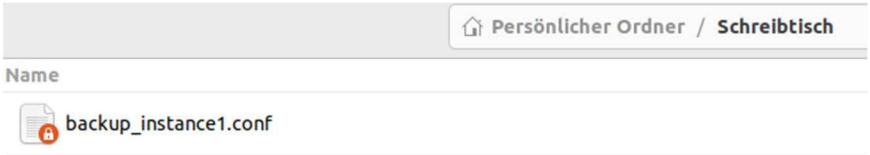

Abbildung 14 Screenshot Konfigurationsdatei RethinkDB

„Welche Funktionen zur Systemsicherung von DBMS und Daten sind vorhanden?“

Hierzu werden jetzt die Frontend und Konfigurationsdateien gesichtet.

Bei CouchDB fällt beispielsweise der „Replikator“ mit seiner Userability auf:

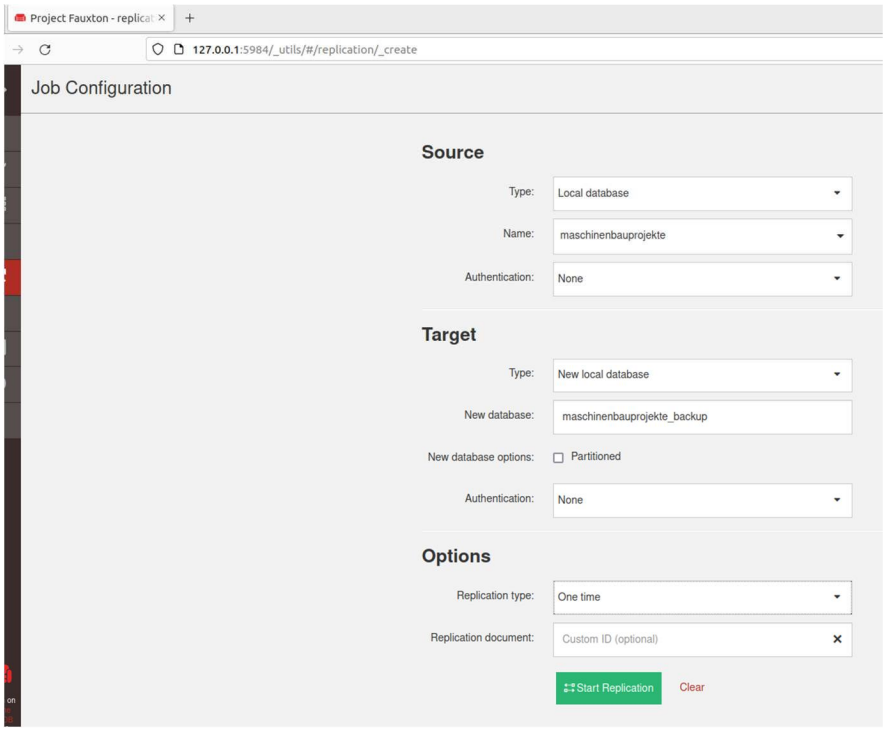

Abbildung 15 Replicator von CouchDB

Aufgrund des grafischen Interfaces muss der Funktionsumfang stärker beschrieben werden. Der Replicator in CouchDB fungiert als Kernkomponente im Prozess der Datenreplikation, indem er die Transferierung von Daten zwischen verschiedenen Instanzen von CouchDB ermöglicht.
Im Hinblick auf die Zugriffskontrolle und Sicherheit während des Replikationsprozesses unterstützt CouchDB die SSL-Verschlüsselung und bietet Authentifizierungsoptionen. Weiterhin stellt CouchDB Protokollierungs- und Überwachungsinstrumente zur Verfügung, die eine Monitoring-Funktionalität des Replikationsstatus und -fortschritts ermöglichen. Im Falle eines Fehlers während der Replikation ist CouchDB in der Lage, den Replikationsprozess bei der nächsten Gelegenheit fortzusetzen, ohne eine Neuintitialisierung des Prozesses zu erfordern, was eine effiziente Fehlerbehandlung und Wiederherstellung unterstützt.
Die Flexibilität wird durch die Option der „Replication type“ unterstrichen, wobei Replikationen entweder manuell oder gemäß einem prädefinierten Zeitplan initiiert werden können. Zusätzlich bietet die Funktion der Filterung die Möglichkeit, die Replikation durch Definition von Filterregeln auf spezifische Dokumente oder Datensätze zu beschränken. Dies ermöglicht eine granulare Kontrolle darüber, welche Daten repliziert werden und welche nicht. Darüber hinaus gibt es weitere externe Tools wie „couchsnap“[33], um eine inkrementelle Sicherung zu realisieren. Da dies ein Community-Tool ist, wurde es für die Bewertung nicht berücksichtigt.

RethinkDB löst Backup und Security klassisch über das Terminal mittels Parameter:

- rethink dump [Parameter]
- -e für Export, um den Dump zu begrenzen
- -f für Datneinamen
- Automatischer Dump zu einem Datum mit Hilfe eines Skripts möglich. Hierzu wird cron genutzt.

Weitere Parameter sind:

- -c, --connect: Host und Client-Port des Knotens, zu dem eine Verbindung hergestellt werden soll (Standard: localhost:28015)

[33]Seite: https://github.com/glynnbird/couchsnap (letzter Aufruf 27.10.2023)

- -p, --password: Aufforderung zur Eingabe des Admin-Passworts, falls eines festgelegt wurde
- --password-file: Das Admin-Passwort aus einer Klartextdatei lesen
- --tls-cert: Geben Sie einen Pfad zu einem TLS-Zertifikat an, um verschlüsselte Verbindungen zum Server zu ermöglichen (siehe Absicherung des Clusters)
- -i, --import: Beschränken Sie die Wiederherstellung auf die angegebene Datenbank oder Tabelle (angegeben als database.table); kann mehrmals für mehrere Datenbanken/Tabellen angegeben werden
- --clients: Anzahl der zu verwendenden Client-Verbindungen (Standard: 8)
- --temp-dir: Verzeichnis für Zwischenergebnisse verwenden
- --hard-durability: Verwenden Sie Hard-Durability-Schreibvorgänge (langsamer, aber weniger Speicherverbrauch auf dem Server)
- --force: Daten importieren, auch wenn bereits eine Tabelle existiert
- --no-secondary-indexes: Keine sekundären Indizes für die wiederhergestellten Tabellen erstellen
- -h, --help: Hilfe anzeigen

3.3 Praktische Herausforderungen beim Bewerten

Wie unter 3.2 Checkliste nach Kavalir (2021)dargestellt müssen anschließend die gefunden Inhalte bewertet werden. An dieser Stelle wird beispielhaft die Auswertung eines spezifischen Aspekts aus der Checkliste vorgestellt. Die vollständigen Ergebnisse im Detail sind im Anhang dokumentiert. Unter 2.3.4 Systembaustein APP.4.3 „Relationale Datenbanken“ S. 18 wurde der Teilbaustein APP.4.3.A9 näher vorgestellt. Die aus diesem abgeleiteten Fragen wurden in eine längere Checkliste oftmals mit weiteren Fragen überführt. Um zu einer Bewertung für den Punkt APP.4.3.A9 zu kommen, wird mittels installierter Software und der Dokumentation der Software versucht, jede dieser Fragen zu beantworten. Bei den Fragen muss teilweise der tatsächliche Funktionsumfang ermittelt werden. Andere Fragen sind simplere Ja-Nein-Fragen. Mittels der Punktetabelle (siehe Abbildung 9) wird nun eine passende Punktzahl vergeben. Aus der vergebenen Punktezahl wird ein Durchschnitt gebildet. Die Frage c) fließt nicht in die Bewertung, dass angenommen wird, dass sie nicht

technisch in einer Nicht-Relationalen Datenbank abgebildet werden kann. Bei manchen Fragen gibt es einen Ermessensspielraum. So ist unklar bei Frage a) welcher Umfang zur Höchstpunktzahl führt. Hier hätte im Rahmen der von Kavalir (2021) die maximale Ausprägung vorher beschrieben werden müssen, um Unklarheiten zu verhindern. Darüber hinaus ist die Skala von 0-2 nicht immer geeignet für eine vernünftige Bewertung. Dies fällt dann auf, wenn Antworten wie folgt ausfallen:

(1) Offiziell gibt es das Feature, z.B. Multi-Factor Authentification
(2) In der Dokumentation des Herstellers gibt es einen Work-Around, der aber nicht zum Standard-Software-Produkt gehört oder
(3) Ein anderes Software-Unternehmen bietet eine kostenlose Lösung an, beispiel IBM mit Data-at-Rest für CouchDB[34]
(4) WICHTIG: Community-Lösungen, die nicht Teil der Dokumentation sind und nicht aus einem vergleichbar mit IBM stammenden Kontext stammen, wurde nicht berücksichtigt.

Eine granulare Skala könnte eine solche Lösung anerkennen. In diesem Fall wurde der Aspekt mit 0 Punkten bewertet, da dieser de facto nicht vorhanden ist und somit nicht zum aktuellen Funktionsumfang der Software gehört.

a) Welche Funktionen zur Systemsicherung von DBMS und Daten sind vorhanden?
b) Wird das Einrichten einer automatischen Systemsicherung unterstützt?
c) Sind die Transaktionen wiederherstellbar?
d) Gibt es eine Funktion zur inkrementellen Sicherung?
e) Welche Parameter können vorgegeben werden?

Methodologisch offen bleibt auch, wenn nur volle Punktzahlen pro Item vergeben werden, wie gerundet werden darf, wenn man den Durchschnitt von mehreren Fragen eines Unterbausteins gebildet hat. Aufrunden würde in diesem Fall bedeuten, dass die Software eigentlich inhaltlich aufgewertet wird, ohne dass damit Inhalte verbunden sind. Aufgrund der Transparenz wurde sich in diesem Fall den ungerundeten Durchschnittswert in Klammern

[34] IBM (2019) „Implement encryption of data at rest in CouchDB Server“

hinter der vergebenen Wertung anzugeben. Dies kam in drei Fällen vor und wird in diesen Fällen auch gesondert erklärt, damit die Wertung transparent und nachvollziehbar erfolgt.

3.3 Auswertung

Die Untersuchung von CouchDB und RethinkDB offenbart diverse Gemeinsamkeiten, jedoch auch signifikante Unterschiede zwischen den beiden Datenbanksystemen. Im Kontext von APP.4.3 – „Relationale Datenbanksysteme“ wies RethinkDB bei der A3 Basishärtung Defizite aufgrund des begrenzten Funktionsumfangs in der Userverwaltung und Authentifizierung auf. Das Fehlen eines geschützten Superadmin-Accounts bei der Installation und das fehlende Rollensystem sind weitere Schwachstellen, die beachtet werden sollten. Die folgenden Ergebnisse wurden beim Systembaustein APP.4.3 – „Relationale Datenbanksysteme“ für die DBMS CouchDB und RethinkDB ermittelt:

- Für A3 Basishärtung erzielten sowohl CouchDB als auch RethinkDB 1 Punkt, wobei für RethinkDB einschränkend folgendes anzumerken ist: Es wurde abgerundet, aufgrund des geringen Funktionsumfangs im Kontext Authentifizierung und Userverwaltung. Darüber hinaus ist anzumerken, dass mit der Installation ein Superadmin-Account ohne Passwortschutz angelegt wird. Zwar lässt sich später ein Passwort hinzufügen, aber eigentlich hätte dieser Punkt negativ bewertet werden müssen. Dazu kommt das fehlende Rollensystem in RethinkDB. Dies verhindert, dass Standards wie „Least Privilege Principle“ implementiert werden können. Da bei RethinkDB fiele Grundlagen fehlen, muss dies vom User sichergestellt werden.
- Bei A9 Datensicherung erzielten beide DBMS 1 Punkt.
- In der Kategorie A13 Handhabung DB-Links konnten sowohl CouchDB als auch RethinkDB die Anforderung nicht erfüllen und erhielten jeweils 0 Punkte.
- Für A16 Verschlüsselung von Verbindungen erzielte CouchDB 2 Punkte, während RethinkDB 1 Punkt erhielt.
- Bei A18 Überwachung erzielte CouchDB 1 Punkt, wohingegen RethinkDB ebenfalls 1 Punkt erhielt. Hier wurde das Ergebnis von 0,5 jedoch nicht zum Anlass abzurunden, sondern es wurde, wie unter **Fehler! Verweisquelle konnte nicht gefunden werden.** (S. **Fehler! Textmarke nicht definiert.**f) angekündigt, aufgerundet. Dies liegt daran,

dass die Art und Weise wie Datensicherung realisiert wurde, zwar besser sein könnte, aber grundlegende Features vorhanden sind.

- Sowohl CouchDB als auch RethinkDB erfüllten A19 Schutz vor Datenbankskripten und erhielten hierfür jeweils 1 Punkt.
- Beim A21 Einsatz von Security Tools erreichten CouchDB 1 Punkt und RethinkDB ebenfalls 1 Punkt.
- Für A24 Data-at-Rest Verschlüsselung erhielt CouchDB 0 Punkte mit einer speziellen Anmerkung von 0,5, während RethinkDB 0 Punkte erhielt. Es gibt seitens IBM ein Konzept samt Anleitung. Dies ist aber kein Standard.

Beim Systembaustein APP.6 – „Allgemeine Software“ wurde nur ein Unterpunkt bewertet:

- In der Kategorie A3 Beschaffung und Integrität erzielten sowohl CouchDB als auch RethinkDB 2 Punkte.

Diese Analyse zeigt auf, in welchem Maße CouchDB und RethinkDB die vorgegebenen Sicherheitsanforderungen erfüllen.

Systembaustein		**CouchDB**	**RethinkDB**
APP.4.3 – „Relationale Datenbanksysteme“	A3 Basishärtung	1	0 (0,6)
	A9 Datensicherung	2	1
	A13 Handhabung DB-Links	0	0
	A16 Verschlüsselung Verbindungen	2	1
	A18 Überwachung	1	1 (0,5)
	A19 Schutz vor Datenbankskripten	1	1
	A21 Einsatz von Security Tools	1	1
	A24 Data-at-Rest Verschlüsselung	0 (0,5)	0
APP.6 – „Allgemeine Software“	A3 Beschaffung und Integrität	2	2

Tabelle 2 Ergebnisse der Analyse

Die Auswertung wurde um die Ergebnisse aus Kavalir (2021)[35] angereichert, so dass ein besserer Vergleich möglich ist. Ein direkter Vergleich mit den Daten aus Kavalir (2021) für MongoDB, OrientDB und Redis zeigt folgendes Bild:
MongoDB und OrientDB übertrafen CouchDB und RethinkDB in der A3 Basishärtung und entsprachen den höchsten Sicherheitsstandards. Redis lag hier im Mittelfeld, vergleichbar mit CouchDB. Interessant ist auch der Vergleich in der Kategorie A16 Verschlüsselung von Verbindungen, in der alle Systeme, mit Ausnahme von RethinkDB, zwei Punkte erzielten. Bei der A18 Überwachung war ein gemischtes Bild zu beobachten. Während CouchDB und Redis jeweils einen Punkt erzielten, blieb OrientDB ohne Punkt. RethinkDBs Bewertung in dieser Kategorie wurde aufgrund der vorhandenen Basisfeatures aufgerundet.

Systembaustein		**MongoDB**	**OrientDB**	**Redis**
APP.4.3 – „Relationale Datenbanksysteme“	A3 Basishärtung	2	2	1
	A9 Datensicherung	1	1	2
	A13 Handhabung DB-Links	0	0	0
	A16 Verschlüsselung Verbindungen	2	2	2
	A18 Überwachung	1	0	1
	A19 Schutz vor Datenbankskripten	1	1	1
	A21 Einsatz von Security Tools	0	1	0
	A24 Data-at-Rest Verschlüsselung	1	2	1
APP.6 – „Allgemeine Software“	A3 Beschaffung und Integrität	2	1	2

Tabelle 3 Ergebnisse von Kavalir (2021)

Zusammenfassend lässt sich festhalten, dass während CouchDB in den meisten Kategorien solide Ergebnisse zeigte, RethinkDB in einigen kritischen Bereichen Defizite aufwies. Hierbei wurde aus der Gesamtpunktzahl bei Kavalir die Punkte rausgerechnet, welche sich auf den

[35] Kavalir (2021) S. 83

Informationsverbund bezogen haben, so dass beide Analysen sich auf die gleichen Aspekte und Inhalte beziehen.

	CouchDB	**RethinkDB**	**MongoDB**	**OrientDB**	**Redis**
Erzielte Punktzahl	10 von 18	7 von 18	10 von 18	10 von 18	10 von 18
Anteil in % an Gesamt	55%	38%	55%	55%	55%

Tabelle 4 Zusammenfassung Ergebnis für Teilanalyse mit Daten aus Kavalir (2021) für MongoDB, OrientDB und Redis[36]

[36] Kavalir (2021) S.85

4. Fazit & Ausblick

4.1 Fazit Analyse

Die systematische Untersuchung und Bewertung von verschiedenen Datenbanksystemen im Kontext ihrer Sicherheitsfeatures liefert ein aufschlussreiches Bild. Auf Basis der Ergebnisse und des Vergleichs in Tabelle 4 zeigt sich, dass CouchDB, MongoDB, OrientDB und Redis jeweils 55% der maximal möglichen Punkte erzielten. Dies spiegelt einen ähnlichen Reifegrad in Bezug auf die betrachteten Sicherheitskriterien wider. RethinkDB hingegen erzielte lediglich 27% der möglichen Punkte, was auf erhebliche Defizite oder Unterschiede in der Umsetzung von Sicherheitsfunktionen hindeutet. RethinkDB niedriger Score im Vergleich zu den anderen Systemen weist auf spezifische Sicherheitslücken. Hier sind seitens der Datenbankadministration gesonderte Maßnahmen notwendig, um Sicherheit herzustellen.

4.2 Fazit Methodologie

Im Rahmen dieser Arbeit wurde deutlich, dass sich die in Kavalir (2021) vorgestellte Checkliste gut überführen lässt. Es konnte systematisch das Vorgehen übertragen werden. Offen geblieben sind die Fragen, ob die Checkliste, die richtigen Punkte und falls ja, diese auch inhaltlich vollständig abbildet. Die abgeleitete Fragestellung für die Checkliste wurde nicht auf Aktualität überprüft.
Die größte Schwäche des Vorgehens ist der Bewertungsmaßstab. Hier wurde deutlich, dass die gewählte Skalierung von 0-2 nicht ausreicht (Vgl. 2.5 BSI-Bewertungsmaßstab in Kavalir (2021) S. 21), um die Software-Features und Zustand der Software optimal zu bewerten. Dies liegt daran, dass mit der Bewertung 1-2 keine ausreichende Differenzierung möglich ist. Mit 1 wird eine „teilweise“ Umsetzung bewertet. Hier ist jedoch der Umfang der Implementierung unklar, da die sprachliche Aussage „teilweise“ zu ungenau ist. Ähnliches gilt auch für die 2. Es macht einen Unterschied, wie ein Feature für Sicherheit implementiert ist. Dies wird beispielsweise daran deutlich, ob ein Grundfeature wie Passwortschutz sofort mit der Installation aktiv für den Admin-Account implementiert wird, wie dies bei CouchDB ist, wo der User mit der Installation ein Passwort setzen muss. Im Gegensatz dazu steht RethinkDB, wo der Admin-Account nicht einmal einen aktiven Standardpasswortschutz besitzt.

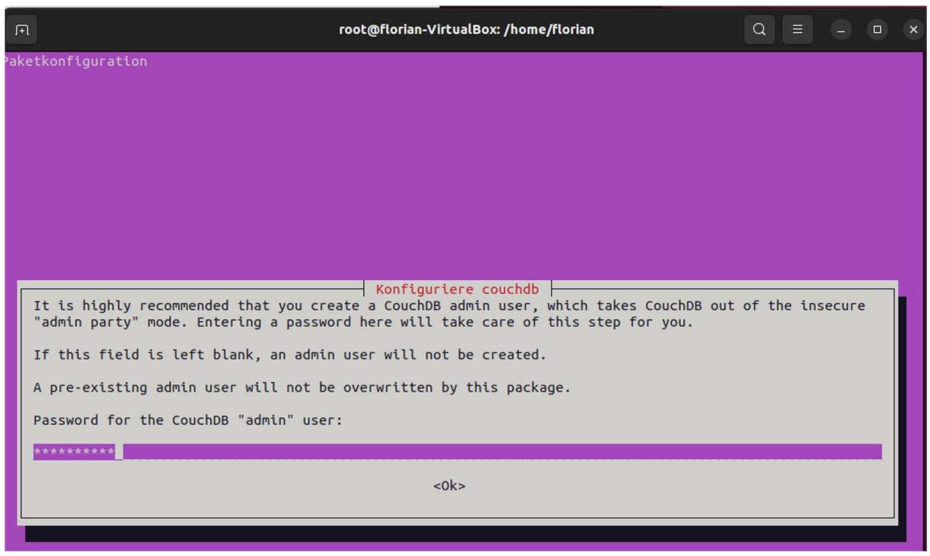

Abbildung 16 Einrichten des Passwortes bei couchDB Installation

Die Tatsache, dass der User für einen Admin-Account nachträglich überhaupt ein Passwort setzen muss, sollte eigentlich zur Abwertung führen. Zumindest wird deutlich, dass ein Feature sehr unterschiedlich implementiert sein kann, wodurch die Sicherheit der Software sich verändert. Dies bildet aber eine Skala von 0-2 nicht ab. Hier stellt sich die Frage, wie muss den ein Feature implementiert sein, damit es aus der Perspektive der Datenbankhärtung die Höchstpunktzahl erzielt. Ein weiterer Kritikpunkt am Bewertungsmaßstab ist das Fehlen von Musterantworten für die Skala von 0-2. Es fehlt gerade bei Fragen nach dem Funktionsumfang eine Beschreibung der Zielversion für das Vergeben der maximalen Punktzahl.

Diese Bewertungsprobleme könnten mit Hilfe eines anderen Bewertungsmaßstabs gelöst werden.

Eine Alternative könnte eine angepasste Skala sein, beispielsweise des „National Defense Information Sharing And Analysis Center“[37] (NDISAC). Die im NDISAC Whitepaper vorgestellte Methode zur Bewertung besteht aus einen sechsstufigen Findungsprozess[38]. Nach dem Bilden von inhaltlichen Kategorien und Kriterien wird dann für diese eine Gewichtung[39] eingeführt.

[37] NDISAC ist ein Teil des Pentagons. Link: https://ndisac.org/ (letzter Aufruf 26.10.2023)
[38] NDISAC (2020) S. 9ff
[39] NDISAC (2020) S.12

Kavalir	Punkte	NDISAC		Punkte
		Product exceeds expectations	Software-Feature übertrifft die Erwartungen: Es ist aktiv und hat einen größeren Funktionsumfang als am Markt üblich	4
Ja	2	Product meets requirement	Software-Feature ist vorhanden und ist aktiv	3
		Product meets requirement sub-optimally	Software-Feature ist vorhanden und ist inaktiv	2
Teilweise	1	Product partially meets requirement	Software-Feature ist vorhanden, muss aber aufwendig nachträglich installiert werden	1
Nein	0	Product does not meet the requirement	Software-Feature ist nicht vorhanden	0

Tabelle 5 Bewertungsmaßstäbe im Vergleich[40]

Später wird das Gewicht mit der erzielten Punktzahl verrechnet werden, so dass die Frage eine am Inhalt gewichtete Bewertung erhält. Die Vergabe der Punkte erfolgt unter bestimmten, mehr detailliert beschriebenen Voraussetzungen. Hierzu kann man entweder für jede Frage und jede Punktewertung eine Musterantwort erstellen. Alternativ wird statt „Ja, Nein, Teilweise“ eine genauere Definition für jedes abgefragtes Item erstellt, die den Rahmen für die Punktevergabe darstellt. In der Tabelle 5 wurde beispielsweise definiert, dass ein Feature vorhanden sein muss und möglichst aktiv implementiert sein soll, damit es möglichst einen hohen Einfluss auf die Sicherheit der Software hat. Nur dann werden 3 Punkte vergeben. 4 Punkte werden nur vergeben, wenn das Feature außergewöhnlich implementiert ist. In Bezug auf die analysierte Passwortimplementierung von CouchDB und RethinkDB würde CouchDB 3 Punkte erzielen („Software-Feature ist vorhanden und aktiv“). Während RethinkDB für die inaktive Implementierung nur 2 erhält („Software-Feature ist vorhanden und ist inaktiv“). Bei der ursprünglichen Bewertung hätten beide Datenbanken 2 Punkte erhalten, da das Feature Passwortschutz implementiert ist.
Durch den neuen Bewertungsmaßstab ist es möglich, Software, welche „Secure by default“[41] implementiert realisiert, besser zu bewerten. Sollte wie in diesen Fall der Passwortschutz auch noch die Sicherheit des Passwortes überprüfen (Länge, Groß- und Kleinschreibung, Zahlen, Zahl der Login-Versuche begrenzt, etc.), da wäre der Standard-Passwortschutz außergewöhnlich gut implementiert, so dass u.U. dies mit 4 Punkten hätte bewertet werden können.[42]

4.3 Ausblick

Die Analyse hat gezeigt, dass es nicht nur eine technisch ausgereifte Checkliste braucht, um mit Hilfe einer Analyse Datenbankhärtung zu bewerten. Wenn es um „Secure by default“ geht, dann muss im Rahmen der Bewertung auch die Art und Weise der Implementierung von Sicherheitsfeatures fokussiert werden (vgl. „

[40] Vgl. NDISAC (2020) S. 13
[41] Faber (2021) S. 40 „Secure by Design bedeutet, dass Software so konzipiert, entwickelt und implementiert sein soll, dass sie sich selbst schützt sowie die Daten, die sie verarbeitet.“
[42] Pohlmann (2022) S. 187ff

“ S. 35). Um die Sicherheit zu erhöhen, müssen aber nicht nur relevante Features aktiv in die Software integriert werden, sondern Sicherheit darf nicht nur ein Thema für Experten zu sein. Die Konfiguration von Software mittels ini-Dateien und Texteditor führt m.E. dazu, dass Menschen von diesen Themen abgeschreckt werden. Diese Themen müssen leichter zugänglich sein, ggf. muss auch das Feature innerhalb der Software besser erläutert werden. Schaut man auf RethinkDB wird beispielsweise deutlich, dass Datenbankhärtung für diese Software irrelevant ist. Dies wird z.B. deutlich bei Thema Usermanagement und Rollen.

```
root@florian-VirtualBox:/home/florian/Dokumente# python3 get_permissions.py

Datenbank: Maschinenbauprojekte
  Benutzer: ['admin'], Berechtigungen: {'config': True, 'connect': True, 'read': True, 'write': True}
  Benutzer: ['admin', 'b99b9572-8e53-5fef-9a2b-5e6aba3edf1c'], Berechtigungen: {'config': True, 'read': True, 'write': True}
  Benutzer: ['nur_lesen'], Berechtigungen: {'config': False, 'connect': False, 'read': True, 'write': False}

Datenbank: rethinkdb
  Benutzer: ['admin'], Berechtigungen: {'config': True, 'connect': True, 'read': True, 'write': True}
  Benutzer: ['admin', 'b99b9572-8e53-5fef-9a2b-5e6aba3edf1c'], Berechtigungen: {'config': True, 'read': True, 'write': True}
  Benutzer: ['nur_lesen'], Berechtigungen: {'config': False, 'connect': False, 'read': True, 'write': False}

Datenbank: test
  Benutzer: ['admin'], Berechtigungen: {'config': True, 'connect': True, 'read': True, 'write': True}
  Benutzer: ['admin', 'b99b9572-8e53-5fef-9a2b-5e6aba3edf1c'], Berechtigungen: {'config': True, 'read': True, 'write': True}
  Benutzer: ['nur_lesen'], Berechtigungen: {'config': False, 'connect': False, 'read': True, 'write': False}
root@florian-VirtualBox:/home/florian/Dokumente#
```

Abbildung 17 Python-Script zur Abfrage von Usern und Rechtevergabe in RethinkDB

Es sind keine Rollen implementiert und es ist auch nicht leicht möglich, User und Rechtevergabe zu monitoren, so dass teilweise nur gute Scripting-Kenntnisse helfen. Dies führt dazu, dass Sicherheit unter Umständen ignoriert wird.

Im Rahmen von Datenbankhärtungsanalysen muss eine solche Implementierung wie bei RethinkDB stärker abgewertet werden. Dafür wäre ein differenzierter Bewertungsrahmen hilfreich. Auch die Erwartung an das abgefragte Item müsste klarer und detaillierter vorab ausformuliert werden. Hier stellt der vorgestellte Bewertungsrahmen der NDISAC eine Möglichkeit, Sicherheit differenzierter zu bewerten.

5. Quellenverzeichnis

(1) Allianz.com (2023) „Allianz Risk Barometer“ Seite: https://www.allianz.com/content/dam/onemarketing/azcom/Allianz_com/press/document/Allianz-Risk-Barometer-2023.pdf (letzter Aufruf 25.10.2023)

(2) Bendig, Henner (2020) „Untersuchung von openscource Datenbanksystemen auf Härtungsmaßnahmen unter Betrachtung des BSI IT-Grundschutz-Kompendium“; Link: (zuletzt besucht 15.10.2023)

(3) Bundesamt für Sicherheit in der Informationstechnik (2023) „IT-Grundschutz Kompendium“ Seite: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2023.pdf?__blob=publicationFile&v=4 (letzter Aufruf 26.10.2023)

(4) Bundesamt für Sicherheit in der Informationstechnik (2023) „IT-Grundschutz-Kompendium – Werkezeug für Informationssicherheit Edition 2023“ Seite: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/it-grundschutz-kompendium_node.html (letzter Aufruf 26.10.2023)

(5) Bundesamt für Sicherheit in der Informationstechnik (2023) „Lerneinheit 2.8: Das Sicherheitskonzept“ Seite: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/Zertifizierte-Informationssicherheit/IT-Grundschutzschulung/Online-Kurs-IT-Grundschutz/Lektion_2_Sicherheitsmanagement/2_08_Sicherheitskonzept.html (letzter Aufruf 26.10.2023)

(6) Bundesamt für Sicherheit in der Informationstechnik (2023) „Lerneinheit 5.2: Schichtenmodell“ Seite: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/Zertifizierte-Informationssicherheit/IT-Grundschutzschulung/Online-Kurs-IT-Grundschutz/Lektion_5_Modellierung/Lektion_5_02/Lektion_5_02_node.html (letzter Aufruf 26.10.2023)

(7) Bundesamt für Sicherheit in der Informationstechnik (2023) „Lerneinheit 6.3: Dokumentation“ Seite: 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/6_03_Dokumentation.html (letzter Aufruf 26.10.2023)

(8) Database of Databases ”dbdb.io” (2018) ”RethinkDB” Seite: https://dbdb.io/db/rethinkdb (letzter Aufruf 26.10.2023)

(9) Faber, Eberhard von (2021) „IT und IT-Sicherheit in Begriffen und Zusammenhängen – Thematisch sortiertes Lexikon mit alphabetischem Register zum Nachschlagen“, Springer Vieweg

(10) Github (2023) „rethinkdb“ Seite: https://github.com/rethinkdb/rethinkdb (letzter Aufruf 26.10.2023)

(11) IBM (2019) „Implement encryption of data at rest in CouchDB Server“ Seite: https://developer.ibm.com/tutorials/implement-custom-solution-kubernetes-cluster-couchdb-ibm-cloud/ (letzter Aufruf: 26.10.2023)

(12) Kaufmann, Michael & Andreas Meier (2023) „SQL- & NoSQL-Datenbanken“, Springer Vieweg, 9. erweiterte und aktualisierte Auflage

(13) Kavalir, Sebastian (2023) „Untersuchung von Nicht-relationalen Datenbanksystemen mit Fokus auf Härtungsmaßnahmen“ Seite: https://it-forensik.fiw.hs-wismar.de/images/b/b2/MT_Kavalir.pdf (letzter Aufruf 26.10.2023)

(14) National Defense-ISAC (2020) „Software Security Automation: Security Controls Evaluation Criteria“ Seite: https://ndisac.org/wp-content/uploads/2020/10/ND-ISAC-Software-Security-Automation-Security-Controls-Evaluation-Criteria-Final_Oct2020.pdf (letzter Aufruf 27.10.2023)

(15) Piller, Ernst (2017) „Beschaffung unter Berücksichtigung der IT Sicherheit – Wichtigkeit, Herausforderungen und Maßnahmen“, Springer Vieweg

(16) Pohlmann, Norbert (2022) „Cyber-Sicherheit – Das Lehrbuch für Konzepte, Prinzipien, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung“, Springer Vieweg, 2. Auflage

(17) Schicker, Edwin (2017) „Datenbanken und SQL – Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL“, Springer Vieweg, 5. Aktualisierte und erweitere Auflage

(18) Statista.com (2023) „Ranking of the most popular database management systems worldwide, as of September 2023" Seite: https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/ (letzter Aufruf 25.10.2023)

(19) Wikipedia (2023) „CouchDB" Seite: https://de.wikipedia.org/wiki/CouchDB (letzter Aufruf 26.10.2023)

(20) Wikipedia (2023) „RethinkDB" Seite: https://de.wikipedia.org/wiki/RethinkDB (letzter Aufruf 26.10.2023)