APP.4.5 Key-Value-Datenbanken

1. Beschreibung

1.1. Einleitung

Key-Value-Datenbanken, oft auch als Key-Value-Stores bezeichnet, sind eine fundamentale Kategorie nicht-relationaler Datenbanken. Systeme wie Redis, Memcached oder RocksDB speichern Daten in einfachen Schlüssel-Werte-Paaren und sind für extrem hohe Lese- und Schreibgeschwindigkeiten optimiert. Häufig werden sie als In-Memory-Datenbanken betrieben, um als hochperformanter Cache, Session-Store oder für Echtzeitanalysen zu dienen.
Ihre Architektur, die auf maximale Performance und geringe Latenz ausgelegt ist, führt oft zu Standardkonfigurationen, die aus Sicherheitssicht problematisch sind. Viele dieser Systeme bieten standardmäßig keine oder nur rudimentäre Authentifizierungsmechanismen, sind im Netzwerk unverschlüsselt erreichbar und verfügen über mächtige Administrationsbefehle, die nicht ausreichend geschützt sind. Diese Eigenschaften stellen spezifische Anforderungen an die Absicherung, die sich grundlegend von denen relationaler Datenbanksysteme unterscheiden.

1.2. Zielsetzung

Dieser Baustein zeigt einen systematischen Weg auf, um Key-Value-Datenbanken sicher zu konfigurieren und zu betreiben. Ziel ist es, die Vertraulichkeit, Integrität und Verfügbarkeit der gespeicherten Daten zu gewährleisten. Dies umfasst die Absicherung des Netzwerkzugriffs, die Etablierung von Authentifizierungs- und Autorisierungsmechanismen, die Härtung der Konfiguration und die sichere Anbindung an die zugehörigen Anwendungen.

1.3. Abgrenzung und Modellierung

Der Baustein APP.4.5 ist auf alle eingesetzten Key-Value-Datenbanken anzuwenden. Dieser Baustein ist eine Ergänzung zum Baustein APP.4.3 Relationale Datenbanksysteme und deckt die spezifischen Anforderungen nicht-relationaler Key-Value-Stores ab. Grundlegende Anforderungen an die Absicherung des darunterliegenden Betriebssystems sind in den Bausteinen der Schicht OPS.1.1 Server zu finden. Die sichere Entwicklung von Anwendungen, die diese Datenbanken nutzen, wird in CON.1 Softwareentwicklung behandelt.

2. Gefährdungslage

Für den Baustein APP.4.5 sind folgende spezifische Bedrohungen und Schwachstellen von besonderer Bedeutung:

Unautorisierter Zugriff auf sensible Daten im Speicher

Da viele Key-Value-Datenbanken standardmäßig ohne Authentifizierung betrieben werden und im Netzwerk lauschen, können Angreifer mit reinem Netzwerkzugriff auf den gesamten Datenbestand zugreifen. Dadurch können sensible Informationen wie Session-Token,

persönliche Daten oder Konfigurationsparameter ausgelesen, manipuliert oder gelöscht werden.

Ausnutzung unsicherer Standardkonfigurationen

Hersteller optimieren Key-Value-Datenbanken oft auf einfache Inbetriebnahme und maximale Leistung, nicht auf Sicherheit. Häufige Schwachstellen in der Standardkonfiguration sind das Binden an alle Netzwerkschnittstellen, das Fehlen einer Passwortabfrage oder die Aktivierung von potenziell gefährlichen Debugging-Funktionen. Werden diese Einstellungen nicht angepasst, sind die Systeme offen für Angriffe.

Manipulation von zwischengespeicherten Daten (Cache Poisoning)

Gelingt es einem Angreifer, Daten im Cache zu manipulieren, kann dies weitreichende Folgen für die angebundene Anwendung haben. Beispielsweise könnten durch die Manipulation von zwischengespeicherten Benutzerrechten unautorisierte Zugriffe ermöglicht oder durch das Einschleusen von Schadcode in gecachte Webseiten-Fragmente Cross-Site-Scripting-Angriffe durchgeführt werden.

Denial-of-Service durch Ressourcenerschöpfung

Da Key-Value-Stores häufig im Arbeitsspeicher (RAM) operieren, sind sie anfällig für Angriffe, die auf eine Erschöpfung der Speicherressourcen abzielen. Ein Angreifer könnte ohne Authentifizierung massenhaft Daten in die Datenbank schreiben, bis der Arbeitsspeicher voll ist und das System oder die davon abhängige Anwendung ausfällt.

Unsichere Kommunikation im Cluster-Betrieb

In einem Cluster-Setup kommunizieren die einzelnen Knoten untereinander, um Daten zu replizieren oder den Cluster-Status abzugleichen. Erfolgt diese Kommunikation unverschlüsselt, können Angreifer im selben Netzwerksegment Daten abhören oder manipulieren. Übernehmen sie einen Knoten, kann dies zur Kompromittierung des gesamten Clusters führen.

3. Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins APP.4.5 aufgeführt.

3.1. Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für diesen Baustein vorrangig erfüllt werden.

APP.4.5.A1 Absicherung des Netzwerkzugriffs (B)

Der Netzwerkzugriff auf die Key-Value-Datenbank MUSS auf das Nötigste beschränkt werden. Die Datenbank SOLLTE nur an eine vertrauenswürdige Netzwerkschnittstelle (z. B. localhost bei lokaler Nutzung) gebunden werden. Zusätzlich MUSS der Zugriff auf Port-Ebene durch eine Firewall auf die Systeme beschränkt werden, die zwingend auf die Datenbank zugreifen müssen.

APP.4.5.A2 Aktivierung einer grundlegenden Authentifizierung (B)

Sofern von der Datenbank unterstützt, MUSS ein Authentifizierungsmechanismus aktiviert werden. Mindestens MUSS ein sicheres Passwort für den Zugriff konfiguriert werden (z. B. über requirepass bei Redis).

APP.4.5.A3 Härtung der Konfiguration (B)

Potenziell gefährliche Befehle, die administrative Eingriffe erlauben (z. B. das Auslesen der gesamten Konfiguration oder das Löschen aller Daten), MÜSSEN deaktiviert oder durch Umbenennung geschützt werden. Unsichere Standardeinstellungen MÜSSEN identifiziert und korrigiert werden.

APP.4.5.A4 Ausführung mit dediziertem Benutzerkonto (B)

Die Key-Value-Datenbank MUSS unter einem eigenen, dedizierten Benutzerkonto mit minimalen Rechten betrieben werden. Der Betrieb unter einem administrativen Konto (z. B. root) ist verboten.

3.2. Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik. Sie SOLLTEN grundsätzlich erfüllt werden.

APP.4.5.A5 Verschlüsselung der Datenübertragung (S)

Die gesamte Kommunikation zwischen den angebundenen Anwendungen und der Key-Value-Datenbank sowie die Kommunikation zwischen den Knoten in einem Cluster SOLLTE mittels TLS verschlüsselt werden.

APP.4.5.A6 Erstellung eines differenzierten Rechtekonzepts (S)

Sofern von der Datenbank unterstützt (z. B. über Redis ACLs), SOLLTE ein differenziertes Rechtekonzept umgesetzt werden. Für jede Anwendung SOLLTE ein eigener Benutzer mit den minimal erforderlichen Berechtigungen für Befehle und Schlüsselmuster erstellt werden.

APP.4.5.A7 Sichere Konfiguration der Datenpersistenz (S)

Falls die Key-Value-Datenbank Daten auf Festplatten persistiert (z. B. für Backups oder Neustarts), SOLLTE sichergestellt werden, dass die erzeugten Dateien mit restriktiven Dateiberechtigungen versehen sind. Der Speicherort SOLLTE so gewählt werden, dass unbefugte Zugriffe verhindert werden.

APP.4.5.A8 Regelmäßige Protokollierung und Überwachung (S)

Alle Verbindungsversuche und ausgeführten Befehle SOLLTEN protokolliert und an ein zentrales Log-Management-System übergeben werden. Die Protokolle SOLLTEN regelmäßig auf verdächtige Aktivitäten, wie fehlgeschlagene Authentifizierungen oder die Nutzung gefährlicher Befehle, ausgewertet werden.

3.3. Anforderungen bei erhöhtem Schutzbedarf

Die folgenden Anforderungen sind exemplarische Vorschläge für ein Schutzniveau, das über den Stand der Technik hinausgeht. Sie SOLLTEN bei erhöhtem Schutzbedarf in Betracht gezogen werden.

APP.4.5.A9 Absicherung der Cluster-Kommunikation mit mTLS (H)

In Cluster-Umgebungen SOLLTE die gesamte Kommunikation zwischen den Knoten (Replikation, Statusabgleich) durch gegenseitige TLS-Authentifizierung (mTLS) abgesichert werden. Jeder Knoten MUSS die Identität der anderen Knoten anhand von Zertifikaten überprüfen.

APP.4.5.A10 Einsatz in einer isolierten und gehärteten Laufzeitumgebung (H)

Die Key-Value-Datenbank SOLLTE in einer dedizierten, gehärteten Laufzeitumgebung betrieben werden, beispielsweise in einem minimalen Container-Image. Der Netzwerkzugriff und die Systemaufrufe (Syscalls) SOLLTEN durch technische Maßnahmen (z. B. AppArmor, seccomp) auf das absolute Minimum beschränkt werden.

APP.4.5.A11 Verschlüsselung von persistenten Daten (Encryption at Rest) (H)

Wenn hochsensible Daten verarbeitet und persistent gespeichert werden, SOLLTEN die Daten auf dem Speichermedium verschlüsselt werden. Dies SOLLTE entweder durch Funktionen des Betriebssystems (z. B. Festplattenverschlüsselung) oder durch datenbankeigene Mechanismen erfolgen.

APP.4.5.A12 Regelmäßige Konfigurations- und Schwachstellen-Audits (H)

Die Konfiguration der Key-Value-Datenbank und die Version der eingesetzten Software SOLLTEN regelmäßig automatisiert auf bekannte Schwachstellen und Abweichungen von einer sicheren Basiskonfiguration (Hardening-Baseline) überprüft werden.