Abstract

In den letzten Jahren haben sich Message Queuing Systeme als essenzielle Komponente moderner verteilter Software-Architekturen etabliert. Sie ermöglichen asynchrone Kommunikation, erhöhen die Zuverlässigkeit von Systemen und unterstützen deren horizontale Skalierbarkeit. Das Federated-Identity-Management-Team bei CANCOM Managed Services GmbH (im Folgenden mit FIM-Team abgekürzt) befindet sich aktuell in einem Umstrukturierungsprozess, in dessen Rahmen eine bestehende Software-Lösung auf den Einsatz einer Message Queue umgestellt wird. Ziel dieser Arbeit ist es, die dabei eingesetzte Technologie zu analysieren, getroffene Entscheidungen sicherheitstechnisch einzuordnen und Handlungsempfehlungen für eine langfristig sichere Architektur abzuleiten.

Zu Beginn werden grundlegende Eigenschaften von Message Queues erläutert und deren Stärken im Vergleich zu direkten Kommunikationsmodellen herausgearbeitet. Anschließend wird die konkrete Architektur des FIM-Teams untersucht, um die getroffenen Sicherheitsmaßnahmen entlang des Entwicklungsprozesses zu bewerten und Empfehlungen auszusprechen. Als Bewertungsrahmen dienen die OWASP Top Ten 2021, ein etablierter Standard zur Identifikation und Priorisierung von Sicherheitsrisiken in Softwaresystemen. Die Analyse zeigt, dass das FIM-Team bereits ein gut durchdachtes Sicherheitskonzept verfolgt und zahlreiche Risiken aus den OWASP Top Ten adressiert. Unter anderem wird der Zugang zur Message-Queue-Infrastruktur architekturbedingt stark eingeschränkt: Externe Nutzer:innen besitzen keinen direkten Zugriff, und interne Mitarbeiter:innen interagieren ausschließlich über abgeschottete Netzwerkbereiche. Zugriffsbeschränkungen werden granular mittels Access Control Lists umgesetzt und Passwörter ausschließlich verschlüsselt übertragen. Darüber hinaus verhindern einheitliche Nachrichtenformate und definierte Verarbeitungsketten die Einspeisung unerlaubter Inhalte, während externe Software-Komponenten bewusst aus vertrauenswürdigen und regelmäßig überprüften Quellen bezogen werden, um Risiken durch kompromittierte Software zu vermeiden.

Trotz des hohen Sicherheitsniveaus wurden Optimierungspotenziale identifiziert. Besonders relevant ist die Einführung einer durchgängigen TLS-Verschlüsselung zur Absicherung der Kommunikation zwischen Systemkomponenten. Aktuell wird dieses Risiko durch eine stark abgeschottete Netzwerkarchitektur kompensiert. Perspektivisch kann eine Migration auf Kubernetes dabei unterstützen, TLS automatisiert auszurollen, ohne die interne Exklusivität des Systems aufzugeben. Außerdem gibt es konfigurationstechnisch leichtes Verbesserungspotenzial. Darüber hinaus empfiehlt es sich, verwendete Software-Versionen regelmäßig zu aktualisieren, um sicherheitsrelevante Verbesserungen zeitnah zu übernehmen. Obwohl die derzeit eingesetzten Versionen nicht veraltet sind, existieren bereits neuere Varianten, deren Nutzung zusätzliche Sicherheit bieten kann. Ergänzend wird für den Produktivbetrieb der Einsatz eines Security-Monitorings empfohlen, etwa durch die Kombination von Loki und Grafana, um sicherheitsrelevante Ereignisse systematisch und visuell nachvollziehen zu können.

Diese Arbeit stellt dem FIM-Team eine strukturierte Übersicht über bestehende und empfohlene Sicherheitsmaßnahmen zur Verfügung. Sie unterstützt die Einordnung neuer Sicherheitsansätze im Kontext der bestehenden Architektur und macht transparent, welche konkreten Risikokategorien durch welche Maßnahmen adressiert werden. Da Software-Sicherheit ein dynamisches Feld darstellt, wird im Ausblick auf die Notwendigkeit kontinuierlicher Sicherheitsanalysen hingewiesen. Zum Zeitpunkt der Erstellung dieser Arbeit wurden die OWASP Top Ten 2025 veröffentlicht, welche neue Risikofaktoren berücksichtigen. Auch insbesondere im Zusammenhang mit dem wachsenden Einsatz von KI-basierten Systemen und Large Language Models kommen stetig neue Gefahren für Softwaresysteme dazu. Dadurch wird deutlich, dass Sicherheitsmaßnahmen stetig an neue technologische Entwicklungen angepasst werden müssen.

Abschließend zeigt sich, dass eine sichere Software-Entwicklung nur durch fortlaufende sicherheitstechnische Überprüfungen gewährleistet werden kann. Die vorliegende Arbeit leistet hierzu einen Beitrag, indem sie bestehende Maßnahmen analysiert, entlang etablierter Standards bewertet und konkrete Handlungsempfehlungen zur weiteren Stärkung der Systemsicherheit formuliert.