FOM Hochschule für Oekonomie & Management

Hochschulzentrum DLS

Exposé

Berufsbegleitender Studiengang
7. Semester
im Studiengang Cyber Security Management (M.Sc.)

im Rahmen der Lehrveranstaltung

über das Thema

Automatisierte Absicherung von Identitäts- und Privilegien-Eskalationsrisiken in Azure Kubernetes Service: Entwicklung und Evaluation eines Policy-as-Code-Frameworks

von

Philipp Rose

Betreuer :
Matrikelnummer : 399061
Abgabedatum : 23. August 2026

Inklusionshinweis

Zur besseren Lesbarkeit wird in dieser Arbeit das generische Maskulinum verwendet. Die in dieser Arbeit verwendeten Personenbezeichnungen beziehen sich – sofern nicht anders kenntlich gemacht – auf alle Geschlechter.

KI-Disclaimer

In dieser Arbeit wurde Künstliche Intelligenz (KI) - in Form von [KI-Tool eintragen] - zur Unterstützung bei der Erstellung und Überarbeitung von Texten eingesetzt. Die KI diente als Hilfsmittel zur Verbesserung der Effizienz und Qualität, wobei alle Inhalte einer sorgfältigen fachlichen Prüfung unterzogen wurden. Entscheidungen über die inhaltliche Ausgestaltung und wissenschaftliche Argumentation wurden ausschließlich vom Autor getroffen. Die Verantwortung für die Richtigkeit und Integrität der Arbeit liegt vollständig beim Verfasser. Der Einsatz von KI erfolgte unter Berücksichtigung ethischer und wissenschaftlicher Standards.

Inhaltsverzeichnis

Abbildungsverzeichnis

Listingverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

1 Einleitung und Titelvorschlag

Azure Kubernetes Service (AKS) übernimmt in produktiven Cloud-Umgebungen zunehmend Aufgaben, bei denen Workload-Identitäten direkten Zugriff auf privilegierte Ressourcen erhalten. Fehlkonfigurationen in Azure Active Directory, etwa bei dynamischen Gruppen oder Managed Identities, ermöglichen dabei nachweislich Privilege-Escalation-Angriffe (Vgl. Bu Haimed, Albahar & Alzubaidi, 2023, Art.-Nr. 226). Innerhalb von Kubernetes-Clustern verschärfen unzureichend abgesicherte Control-Plane-Komponenten und Zugriffskontrollen dieses Risiko zusätzlich (Vgl. Morić, Dakić & Čavala, 2025, Art.-Nr. 30). Das Bedrohungsmodell STRIDE systematisiert Rechteausweitung als eigenständige Bedrohungskategorie (Vgl. Lee & Lee, 2022, S. 1047–1059), das ergänzende Projekt MITRE ATT&CK for Containers katalogisiert containerspezifische Angriffstechniken entlang einer Kill Chain (Vgl. MITRE Center for Threat-Informed Defense, 2021). Automatisierte, durchsetzbare Regeln, die diese Eskalationspfade auf Basis eines festgelegten Bedrohungsmodells erkennen und unterbinden, werden in der einschlägigen Literatur bislang nur für Teilaspekte und ohne durchgängige Kopplung an STRIDE oder MITRE ATT&CK beschrieben.

Bestehende Arbeiten adressieren Teilaspekte dieses Problems, ohne die Kombination aus Identitäts- und Privilegien-Eskalationsrisiken in AKS durchgängig abzudecken. Zero-Trust-Konzepte für Kubernetes-Cluster bleiben in der Regel auf allgemeine Architekturprinzipien beschränkt (Vgl. dos Santos, 2025, S. 57–71) und lassen sich nur bedingt auf die Azure-spezifische Identitätsverwaltung übertragen (Vgl. Dakić et al., 2024, Art.-Nr. 2). Eine gemeinsame Policy-as-Code-Definitionsschicht über CI/CD und Kubernetes-Admission-Control hinweg deckt Identitätsrisiken ihrerseits nicht ab (Vgl. Nogueira & Resende, 2026, Art.-Nr. 453), und ein Compliance-Assessment des Kubernetes Control Plane bleibt ohne automatisierte Regeldurchsetzung (Vgl. Morić, Dakić & Čavala, 2025, Art.-Nr. 30). Dieses Vier-Quellen-Bündel deckt bereits eigenständig die zentralen Teildimensionen des Themenfelds ab. Keine der vier Arbeiten vereint dabei AKS-Fokus, Identitätsbezug und Policy-as-Code-Umsetzung gemeinsam in einem Ansatz. Ergänzend dazu, aber nicht als notwendige Voraussetzung dieser Abgrenzung, wurde ein Framework mit thematisch naheliegendem Fokus auf föderierte Identitäten und Laufzeitüberwachung in AKS auf einer kleineren, regionalen Studierendenkonferenz vorgestellt (Vgl. Kripa, 2025). Es liefert einen zusätzlichen Praxisnähe-Beleg für die Umsetzbarkeit eines solchen Ansatzes, ohne dass die Neuheitsbegründung der vorliegenden Arbeit von dieser Einzelquelle abhängt.

Gegenüber diesem Vier-Quellen-Bündel begründet sich die vorliegende Masterarbeit nicht über eine grundsätzlich neue Fragestellung, sondern über eine methodisch-operationale

Verengung: eine auf genau zwei vollständig konstruierte, demonstrierte und evaluierte Bedrohungen begrenzte Analyse, eine reproduzierbare Übersetzungskette von Bedrohungsmodell zu Regelwerk sowie eine daran gekoppelte Wirksamkeitsmessung. Diese Verengung von Bedrohungsanalyse, Identitätskontrolle und automatisierter Regeldurchsetzung bildet den Ausgangspunkt der Masterarbeit.

Für die Arbeit ergeben sich auf dieser Grundlage ein Hauptvorschlag und zwei alternative Titelformulierungen mit unterschiedlicher Schwerpunktsetzung:

- **Hauptvorschlag:** *Automatisierte Absicherung von Identitäts- und Privilegien-Eskalationsrisiken in Azure Kubernetes Service: Entwicklung und Evaluation eines Policy-as-Code-Frameworks*
- **Alternative mit Zero-Trust-Schwerpunkt:** *Zero-Trust-Absicherung von Workload-Identitäten in Azure Kubernetes Service: Entwicklung eines Policy-as-Code-Frameworks gegen Privilegien-Eskalation*
- **Alternative mit Policy-as-Code-Schwerpunkt:** *Policy-as-Code gegen Privilegien-Eskalation in Kubernetes: Ein bedrohungsmodellbasiertes Framework für Azure Kubernetes Service*

2 Forschungsfrage und Zielsetzung der Arbeit

Die Arbeit widmet sich der Frage, wie sich ein aus STRIDE- und MITRE-ATT&CK-Bedrohungsmodellen abgeleitetes Policy-as-Code-Framework gestalten lässt, das eine feste, priorisierte Auswahl von Identitäts- und Privilegien-Eskalationsrisiken in Azure Kubernetes Service automatisiert erkennt und unterbindet, und wie sich dessen Wirksamkeit messen lässt. Diese Hauptforschungsfrage verbindet die Konstruktion eines Artefakts mit dessen empirischer Überprüfung und bildet den Kern des Design-Science-Research-Vorhabens. Zur Beantwortung werden zwei Unterfragen herangezogen, die Artefaktdesign und Artefaktevaluation als methodisch getrennte Schritte abbilden:

1. Wie lassen sich genau zwei, anhand einer Risikomatrix aus Eintrittswahrscheinlichkeit und Auswirkung priorisierte Identitäts- und Privilegien-Eskalationsbedrohungen nach STRIDE und MITRE ATT&CK for Containers in ein darauf zugeschnittenes Regelwerk (Azure-Policy-Add-on für AKS/OPA Gatekeeper) und in gezielte Identitätskontrollen (Microsoft Entra Workload ID) übersetzen?

2. Wie wirksam ist das entwickelte Framework, gemessen an einem auf dieselben zwei Bedrohungen bezogenen CIS-Benchmark-Compliance-Delta und an zwei entsprechend zugeordneten, mit Stratus Red Team simulierten Angriffsszenarien (Breach & Attack Emulation)?

Ziel der Arbeit ist die Entwicklung und Evaluation eines Policy-as-Code-Frameworks, das zwei priorisierte, in Kubernetes-Umgebungen dokumentierte Rechteausweitungspfade (Vgl. Lee & Lee, 2022, S. 1047–1059) durch automatisierte Regeln und Identitätskontrollen adressiert. Das Artefakt operationalisiert damit eine eng geführte Auswahl aus Identitäts- und Privilegien-Eskalationsrisiken, die bestehende Zero-Trust-Ansätze für Kubernetes (Vgl. dos Santos, 2025, S. 57–71) und rein policy-zentrierte Governance-Konzepte (Vgl. Nogueira & Resende, 2026, Art.-Nr. 453) jeweils nur teilweise abdecken. Der damit verbundene Wissensbeitrag liegt auf der Ebene einer situierten Artefakt-Instanziierung für diesen spezifischen Anwendungsfall (Vgl. Hevner et al., 2004, S. 75–105), nicht auf der Ebene generalisierbarer Design-Prinzipien oder einer eigenständigen Design-Theorie. Die zweite Unterfrage sichert die empirische Überprüfung des Artefakts anhand konkreter, auf dieselben zwei Bedrohungen bezogener Kennzahlen ab.

Die Priorisierung der zwei Bedrohungen erfolgt über eine Risikomatrix, die Eintrittswahrscheinlichkeit und Auswirkung jeweils dreistufig in gering, mittel und hoch einstuft und vorrangig Bedrohungen mit der höchsten kombinierten Einstufung berücksichtigt. Wirksamkeit wird dabei als kombiniertes Kriterium aus zwei Teilergebnissen verstanden, die für

jede der beiden priorisierten Bedrohungen einzeln erhoben werden: einer nachweisbaren Verbesserung des Compliance-Delta gegenüber dem CIS-Benchmark und einer präventiven Abwehr des zugeordneten, mit Stratus Red Team simulierten Angriffsszenarios. Beide Teilkriterien gelten für beide Bedrohungen gleichermaßen. Compliance-Delta und Angriffsabwehr werden dabei jeweils eigenständig erhoben, ein verbessertes Compliance-Delta gleicht eine ausbleibende Angriffsabwehr somit nicht aus und umgekehrt.

Compliance-Delta und Angriffsabwehr beruhen zwar auf denselben Regeln des Azure-Policy-Add-on für AKS, erfassen damit aber unterschiedliche Eigenschaften dieser Regeln. Das Compliance-Delta misst den Konfigurationszustand, also ob die erforderlichen Regeln vorhanden und gegenüber dem CIS-Benchmark korrekt konfiguriert sind. Die Angriffsabwehr misst dagegen das Verhalten dieser Regeln unter einem realen, mit Stratus Red Team ausgeführten Zugriffsversuch und kann von der Konfigurationsprüfung abweichen, etwa wenn eine formal CIS-konforme Regel eine bestimmte Angriffstechnik durch eine Ausnahmedefinition oder eine unvollständige Musterabdeckung dennoch nicht blockiert. Zwischen beiden Kriterien bleibt eine gewisse Korrelation bestehen, da beide auf derselben Regelbasis aufsetzen. Diese Korrelation gilt als bewusst akzeptierte Eigenschaft des Messdesigns, nicht als Mangel, weil beide Kriterien trotz gemeinsamer Regelbasis unterschiedliche Fehlerarten offenlegen, eine konfigurationsseitige Lücke einerseits und eine funktionale Wirksamkeitslücke andererseits.

Das Framework gilt insgesamt als wirksam, wenn für beide Bedrohungen sowohl das Compliance-Delta eine Verbesserung ausweist als auch das jeweils zugeordnete Angriffsszenario als abgewehrt eingestuft wird. Wird stattdessen mindestens ein Angriffsszenario nur als teilweise abgewehrt eingestuft, weil der zugrunde liegende Zugriffsversuch zwar erkannt, aber nicht präventiv verhindert wird, gilt das Framework als teilweise wirksam, sofern das Compliance-Delta für beide Bedrohungen dennoch eine Verbesserung ausweist. Bleibt für mindestens eine Bedrohung entweder die Compliance-Delta-Verbesserung aus oder wird das zugeordnete Angriffsszenario als nicht abgewehrt eingestuft, gilt das Framework insgesamt als nicht wirksam.

Eine Attributionsregel unterscheidet dabei ein inhaltliches Scheitern des Artefaktdesigns von einer extern verursachten Ausführungseinschränkung. Als inhaltliches Scheitern gilt nur der Fall, dass eine Demonstration oder Messung tatsächlich vollständig durchgeführt wurde und das Ergebnis hinter der Erfolgsschwelle zurückbleibt. Kann eine der beiden Bedrohungen dagegen nicht vollständig demonstriert oder gemessen werden, weil eine externe, infrastruktur-, werkzeug- oder budgetbedingte Voraussetzung ausfällt, wird dieser Ausfall als Ausführungseinschränkung dokumentiert und von einem inhaltlichen Scheitern des Artefaktdesigns unterschieden. Die Voraussetzung, dass eine Einstufung als wirksam

oder teilweise wirksam die vollständige Demonstration und Messung für beide priorisierten Bedrohungen erfordert, bleibt davon unberührt.

3 Methodik und Vorgehen

Das methodische Vorgehen der geplanten Masterarbeit folgt dem Design-Science-Research-Paradigma nach Hevner et al., das Forschung in einen Relevance Cycle, einen Rigor Cycle und einen zentralen Design Cycle gliedert (Vgl. Hevner et al., 2004, S. 75–105). Der Relevance Cycle verbindet die in Azure Kubernetes Service beobachteten Identitäts- und Eskalationsrisiken mit der Forschung, der Rigor Cycle bindet bestehendes Wissen zu Bedrohungsmodellen und Policy-as-Code ein. Die methodische Abgrenzung gestaltungsorientierter Forschung gegenüber erklärender Natural-Science-Forschung nach March/Smith ordnet das zu entwickelnde Framework als Artefakt im Sinne der Design Science ein (Vgl. March & Smith, 1995, S. 251–266). Als konkretes Prozessmodell dient die Design Science Research Methodology (DSRM) nach Peffers et al. mit den sechs Phasen Problem Identification, Objectives, Design and Development, Demonstration, Evaluation und Communication (Vgl. Peffers et al., 2007, S. 45–77).

Im Sinne der von Hevner et al. unterschiedenen Artefakttypen Konstrukte, Modelle, Methoden und Instanziierungen (Vgl. Hevner et al., 2004, S. 75–105) ordnet sich das zu entwickelnde Framework als situierte Instanziierung für den konkreten Anwendungsfall der Identitäts- und Privilegien-Eskalation in Azure Kubernetes Service ein. Die Arbeit erhebt damit keinen Anspruch auf generalisierbare Design-Prinzipien oder eine von diesem Anwendungsfall losgelöste Design-Theorie, sondern weist die Umsetzbarkeit und Wirksamkeit der abgeleiteten Regeln für den untersuchten, auf zwei priorisierte Bedrohungen begrenzten Ausschnitt nach.

Die ersten beiden DSRM-Phasen werden durch eine Bedrohungsanalyse von AKS-Workloads operationalisiert. STRIDE dient als Klassifikationsschema für Bedrohungskategorien wie Spoofing, Tampering und Elevation of Privilege (Vgl. Lee & Lee, 2022, S. 1047–1059), MITRE ATT&CK for Containers ergänzt containerspezifische Angriffstechniken entlang der Kill Chain (Vgl. MITRE Center for Threat-Informed Defense, 2021). Die vorausgegangene Bedrohungsmatrix für Kubernetes von Microsoft Defender for Cloud bildet die konzeptionelle Grundlage dieses Vorgehens (Vgl. Weizman, 2020). Aus der Verschneidung beider Bedrohungsmodelle werden genau zwei Identitäts- und Privilegien-Eskalationsbedrohungen ausgewählt, priorisiert über eine Risikomatrix aus Eintrittswahrscheinlichkeit und Auswirkung, die anschließend in Anforderungen an das zu entwickelnde Artefakt überführt werden.

Für die Priorisierung wird eine dreistufige Risikomatrix verwendet, die Eintrittswahrscheinlichkeit und Auswirkung jeweils in den Ausprägungen gering, mittel und hoch bewertet.

Priorisiert werden vorrangig Bedrohungen im Feld hoch/hoch, ergänzt um die nächsthöchste Kombination, falls weniger als zwei Bedrohungen die Höchstausprägung erreichen. Die Einstufung im Feld hoch/hoch stützt sich nicht auf eine eigene CVE-Statistik oder Expertenbefragung, deren Erhebung im Rahmen eines Exposés nicht zu leisten ist, sondern auf die Häufigkeit, mit der Rechteausweitung über Workload-Identitäten in der bereits herangezogenen Literatur zu Kubernetes- und AKS-Bedrohungen als eigenständige Risikokategorie behandelt wird (Vgl. Bu Haimed, Albahar & Alzubaidi, 2023, Art.-Nr. 226)(Vgl. Lee & Lee, 2022, S. 1047–1059)(Vgl. MITRE Center for Threat-Informed Defense, 2021). Diese wiederholte Nennung über drei methodisch unabhängige Quellen hinweg dient als Näherungswert für die Praxisrelevanz der zwei Bedrohungen, nicht als quantitative Risikokennzahl. Auf dieser Grundlage werden Token-Diebstahl bei Workload-Identitäten und Privilege Escalation über fehlkonfigurierte Managed Identities (Vgl. Bu Haimed, Albahar & Alzubaidi, 2023, Art.-Nr. 226) als die zwei priorisierten Bedrohungen festgelegt.

Bei der Zuordnung zu STRIDE ist zwischen den zwei Bedrohungen zu differenzieren. Token-Diebstahl bei Workload-Identitäten stellt primär eine Spoofing- und Information-Disclosure-Bedrohung dar, da ein entwendetes Token die Identität eines legitimen Workloads vortäuscht und dessen Berechtigungen offenlegt, wodurch eine nachgelagerte Rechteausweitung erst ermöglicht, aber nicht selbst dargestellt wird. Privilege Escalation über fehlkonfigurierte Managed Identities lässt sich dagegen direkt der STRIDE-Kategorie Elevation of Privilege zuordnen. Nach MITRE ATT&CK for Containers fällt Token-Diebstahl vorrangig unter die Taktik Credential Access, Privilege Escalation über fehlkonfigurierte Managed Identities unter die Taktik Privilege Escalation (Vgl. MITRE Center for Threat-Informed Defense, 2021). Policy-Regelwerk, Identitätskontrollen, Angriffssimulation und Compliance-Delta-Messung bleiben trotz dieser differenzierten Zuordnung durchgängig an dieselbe, aus STRIDE und MITRE ATT&CK abgeleitete Bedrohungsbasis gekoppelt.

Das Artefakt besteht aus einem Policy-as-Code-Framework auf Basis von Microsoft Entra Workload ID für die Identitätsanbindung von Workloads sowie dem Azure-Policy-Add-on für AKS für die Durchsetzung von Zulassungsregeln im Kubernetes Admission Control. Das Azure-Policy-Add-on setzt technisch selbst auf OPA Gatekeeper auf und bildet damit eine einzige, in sich geschlossene Durchsetzungsebene statt zweier getrennter Policy-Engines. Microsoft Defender for Containers und Defender for Cloud Security Posture Management übernehmen dabei keine eigenständige Konzeptionsebene, sondern dienen als unterstützende Nachweisfunktion zur Verifikation der Angriffserkennung während der Demonstration. Eine vergleichbare Kombination aus Policy-Engine und Admission-Control-Durchsetzung hat sich bereits in einem produktiven Kubernetes-Cluster bewährt (Vgl. Ali et al., 2024, Art.-Nr. 07026), ebenso die Wiederverwendung einer gemeinsamen Policy-as-Code-Definitionsschicht über CI/CD und Admission Control hinweg (Vgl. Nogueira & Re-

sende, 2026, Art.-Nr. 453). Die Konstruktion orientiert sich zusätzlich an der NIST-Leitlinie zur Container-Sicherheit (Vgl. Souppaya, Morello & Scarfone, 2017). Als weitere Zielstandards dienen der CIS Microsoft AKS Benchmark (Vgl. Center for Internet Security (CIS), 2026) sowie der orchestrierungsspezifische BSI-Baustein APP.4.4 Kubernetes (Vgl. Bundesamt für Sicherheit in der Informationstechnik (BSI), 2023).

Da kein Praxispartner zur Verfügung steht, dient eine bewusst minimal gehaltene, auf die zwei priorisierten Bedrohungen zugeschnittene AKS-Referenzarchitektur als Demonstrations- und Evaluationsumgebung, wodurch sich Konstruktion und erste Funktionstests innerhalb des dafür vorgesehenen Zeitfensters von Design and Development durchführen lassen. Die Begrenzung auf zwei vollständig zu konstruierende und zu demonstrierende Bedrohungen reduziert das Zeitrisiko der Einarbeitung in Rego, Open Policy Agent und Microsoft Entra Workload ID gegenüber einem breiteren Bedrohungsumfang, ohne dass dafür ein gesondertes Minimalziel oder eine Rückfalloption erforderlich wäre.

Innerhalb eines Lab-Proof-of-Concept werden zwei Angriffsszenarien simuliert, die den priorisierten Bedrohungen 1:1 zugeordnet sind: Token-Diebstahl und Privilege Escalation über Managed Identities. Für die Breach & Attack Emulation kommt Stratus Red Team zum Einsatz, ein Open-Source-Werkzeug, das Azure- und AKS-spezifische Angriffstechniken MITRE-ATT&CK-orientiert nachbildet. Sollten für eine der zwei priorisierten Bedrohungen keine passenden vordefinierten Stratus-Red-Team-Techniken verfügbar sein, wird ergänzend ein eigenes, an der jeweiligen Bedrohung orientiertes Simulationsskript entwickelt, um die 1:1-Zuordnung von Bedrohung und Angriffsszenario in jedem Fall einzuhalten.

Die Testumgebung wird über das Studierendenkontingent Azure for Students finanziert, was den Kostenrahmen des Proof-of-Concept auf einen niedrigen dreistelligen Betrag begrenzt. Da Microsoft Defender for Containers und Defender for Cloud Security Posture Management laufend abgerechnete Dienste sind, wird die Referenzumgebung zwischen den Testphasen pausiert beziehungsweise abgebaut, um das Guthaben über den gesamten Demonstrations- und Evaluationszeitraum zu strecken. Auf- und Abbau der pausierten Referenzumgebung erfolgen reproduzierbar über Bicep-Vorlagen (Azure Resource Manager), sodass sich die minimale AKS-Referenzarchitektur zwischen den Testphasen konsistent und ohne manuellen Konfigurationsaufwand wiederherstellen lässt.

Die Wirksamkeit des Artefakts wird anhand eines auf die zwei priorisierten Bedrohungen bezogenen Compliance-Delta gegenüber dem CIS-Benchmark sowie anhand der Ergebnisse der zwei zugeordneten Breach-&-Attack-Emulation-Szenarien gemessen. Die Wahl der Evaluationsstrategie folgt dem FEDS-Framework (Framework for Evaluation in Design Science Research) nach Venable et al., das zwischen künstlicher und naturalistischer Evaluation unterscheidet (Vgl. Venable, Pries-Heje & Baskerville, 2016, S. 77–89). Da im

Lab-Proof-of-Concept primär die technische Funktionsfähigkeit des Artefakts im Vordergrund steht und keine menschlichen Stakeholder eingebunden sind, orientiert sich die Evaluation an der von Venable et al. beschriebenen Strategie „Technical Risk & Efficacy“ (Venable, Pries-Heje & Baskerville, 2016, S. 77–89). Damit schließt der Design Cycle mit einer Prüfung, die auf den zuvor abgeleiteten Anforderungen und der Zielsetzung der Arbeit aufbaut.

Ein Angriffsszenario gilt als abgewehrt, wenn das Azure-Policy-Add-on den zugrunde liegenden Zugriffsversuch bereits präventiv über die Admission Control blockiert. Wird ein Zugriffsversuch nicht präventiv verhindert, aber von Microsoft Defender for Containers beziehungsweise Defender for Cloud Security Posture Management zuverlässig als Angriff erkannt, gilt das Szenario als teilweise abgewehrt, da die detektive Nachweisfunktion die Sichtbarkeit des Angriffs sicherstellt, die Verhinderung des Zugriffs jedoch offenbleibt. Bleibt ein Zugriffsversuch sowohl unerkannt als auch unverhindert, gilt das Szenario als nicht abgewehrt.

Compliance-Delta und Angriffsabwehr beruhen zwar auf denselben Zulassungsregeln des Azure-Policy-Add-on im Kubernetes Admission Control, erfassen dabei aber unterschiedliche Eigenschaften dieser Regeln. Das Compliance-Delta misst den Konfigurationszustand gegenüber dem CIS-Benchmark, also ob die für die jeweilige Bedrohung erforderlichen Regeln vorhanden und korrekt konfiguriert sind. Die Angriffsabwehr misst demgegenüber das tatsächliche Verhalten derselben Regeln unter einem realen, mit Stratus Red Team ausgeführten Zugriffsversuch und kann von der Konfigurationsprüfung abweichen, etwa wenn eine gegenüber dem CIS-Benchmark korrekt konfigurierte Regel eine konkrete Angriffstechnik dennoch nicht blockiert, weil eine Ausnahmedefinition greift oder das Muster der Technik von der Regel nicht erfasst wird. Zwischen beiden Kriterien bleibt eine gewisse Korrelation bestehen, da beide auf derselben Regelbasis aufsetzen. Diese Korrelation wird als bewusst akzeptierte Eigenschaft des Messdesigns behandelt, nicht als Mangel, weil Compliance-Delta und Angriffsabwehr trotz gemeinsamer Regelbasis unterschiedliche Fehlerarten offenlegen, eine konfigurationsseitige Lücke einerseits und eine funktionale Wirksamkeitslücke andererseits.

Das Artefakt gilt insgesamt als wirksam, wenn für beide priorisierten Bedrohungen sowohl das Compliance-Delta gegenüber dem zum Zeitpunkt der Evaluation fixierten und dokumentierten CIS-Benchmark eine Verbesserung ausweist als auch das jeweils zugeordnete Angriffsszenario als abgewehrt eingestuft wird. Ein verbessertes Compliance-Delta gleicht dabei eine ausbleibende Angriffsabwehr nicht aus und umgekehrt. Wird mindestens ein Angriffsszenario nur als teilweise abgewehrt eingestuft, während das Compliance-Delta für beide Bedrohungen dennoch eine Verbesserung ausweist, gilt das Artefakt insge-

samt als teilweise wirksam. Bleibt für mindestens eine Bedrohung die Compliance-Delta-Verbesserung aus oder wird ein Angriffsszenario als nicht abgewehrt eingestuft, gilt das Artefakt insgesamt als nicht wirksam.

Eine Attributionsregel unterscheidet dabei, ob eine ausbleibende Einstufung als wirksam auf ein inhaltliches Scheitern des Artefaktdesigns oder auf eine extern verursachte Ausführungseinschränkung zurückgeht. Als inhaltliches Scheitern gilt ausschließlich der Fall, dass eine Demonstration oder Messung für eine der zwei priorisierten Bedrohungen vollständig durchgeführt wurde und das Ergebnis hinter der Erfolgsschwelle zurückbleibt, etwa weil das Azure-Policy-Add-on einen Zugriffsversuch nicht blockiert oder das Compliance-Delta ausbleibt. Kann eine Demonstration oder Messung dagegen nicht vollständig durchgeführt werden, weil eine externe Voraussetzung ausfällt, etwa mangels einer passenden, auch durch ein ergänzend entwickeltes Simulationsskript nicht abbildbaren Stratus-Red-Team-Technik, durch eine vorzeitige Erschöpfung des Guthabens trotz des vorgesehenen Budget-Checkpoints oder durch eine Inkompatibilität eines eingesetzten Azure-Dienstes, wird dieser Fall im Ergebnisteil gesondert als Ausführungseinschränkung dokumentiert. An der Grundregel, dass eine Einstufung als wirksam oder teilweise wirksam die vollständige Demonstration und Messung für beide priorisierten Bedrohungen voraussetzt, ändert die Attributionsregel nichts.

4 Literaturrecherche

Die Literaturrecherche verwendet eine kombinierte Stichwortsuche in arXiv (native Bool'sche Suche über die offizielle arXiv-API) und Google Scholar (Stichwortkombination, Sichtung der ersten 20 Treffer je Cluster). Die Suchbegriffe verknüpfen drei Themenachsen: Kubernetes-Sicherheit („Kubernetes security", „Azure Kubernetes Service"), Identität und Privilegien-Eskalation („privilege escalation", „workload identity", „Azure Active Directory") sowie Policy-Durchsetzung („policy as code", „OPA Gatekeeper", „admission control"). Eingeschlossen werden Veröffentlichungen ab 2020, um die seit der Einführung von MITRE ATT&CK for Containers etablierte Bedrohungsterminologie abzudecken (Vgl. MITRE Center for Threat-Informed Defense, 2021). Vorrangig berücksichtigt werden begutachtete Journal- und Konferenzbeiträge, ergänzt um einschlägige Industriestandards und technische Reports, sofern sie methodisch nachvollziehbare Aussagen zu Bedrohungen oder Gegenmaßnahmen treffen.

Tabelle 1 dokumentiert die drei Suchcluster mit den tatsächlichen Trefferzahlen. Die arXiv-Trefferzahlen stammen aus der offiziellen `opensearch:totalResults`-Angabe der arXiv-API und sind damit exakt, die Google-Scholar-Werte beziehen sich auf die gesichteten ersten 20 Treffer je Suche, da Google Scholar selbst keine exakte Gesamttrefferzahl ausweist. Die geringen arXiv-Trefferzahlen bei Cluster 2 und 3 bestätigen, dass Literatur zur Kombination aus Identität, Privilegien-Eskalation und Policy-Durchsetzung in AKS überwiegend außerhalb von Preprint-Servern erscheint, in Konferenzbänden und Zeitschriften, die über Google Scholar besser abgedeckt sind. Alle vier bereits verwendeten Kernquellen (Shamim et al. 2020, dos Santos 2025, Nogueira/Resende 2026, Morić et al. 2025) tauchen unabhängig in den jeweils passenden Clustern wieder auf, was die Literaturbasis extern bestätigt.

Ergänzend liefern umfassende Survey-Arbeiten zur Bedrohungsmodellierung von Containern (Vgl. Wong et al., 2021) sowie zur Systematisierung von Kubernetes-Sicherheitspraktiken wie Role-Based Access Control und dem Prinzip der geringsten Rechtevergabe (Vgl. Shamim, Bhuiyan & Rahman, 2020) den Grundlagenrahmen, auf dem die vorgesehene Bedrohungsanalyse im Grundlagenkapitel der Masterarbeit aufbaut. Beide Arbeiten benennen Rechteausweitung als wiederkehrendes Risiko, ohne selbst ein automatisiertes Gegenmaßnahmen-Framework vorzuschlagen.

Tabelle 2 ordnet die zentralen verwandten Arbeiten entlang der Dimensionen AKS-/Kubernetes-Fokus, Identitätsbezug, Policy-as-Code-Umsetzung sowie systematischer Zuordnung zu einem Bedrohungsmodell (STRIDE/MITRE ATT&CK) ein.

Tabelle 1: Suchstrategie mit Trefferzahlen je Themencluster

Themencluster	Suchstring	Treffer	Relevant
Kubernetes-/AKS-Sicherheit	„Kubernetes security“ OR „Azure Kubernetes Service“	arXiv 7 / GS 20 gesichtet	9, davon 2 bereits in Literaturbasis (Shamim et al. 2020, Morić et al. 2025)
Identität und Privilegien-Eskalation	„privilege escalation“ AND („workload identity“ OR „Azure Active Directory“)	arXiv 0 / GS 20 gesichtet	6, davon 1 bereits in Literaturbasis (dos Santos 2025)
Policy-Durchsetzung	„policy as code“ AND („OPA Gatekeeper“ OR „admission control“)	arXiv 1 / GS 20 gesichtet	7, davon 1 bereits in Literaturbasis (Nogueira/Resende 2026)

Tabelle 2: Verwandte Arbeiten im Themenfeld AKS-Identität, Privilegien-Eskalation und Policy-as-Code

Quelle	AKS-/Kubernetes-Fokus	Identitätsbezug	Policy-as-Code	Bedrohungsmodell-Zuordnung	Abgrenzung zur eigenen Arbeit
Kripa (2025)	AKS	Föderierte Identität	OPA/Gatekeeper	Nicht erkennbar	Regionale Studierendenkonferenz (Praxisnähe-Beleg statt wissenschaftlicher Ankerpunkt)
dos Santos (2025)	Kubernetes allgemein	Zero-Trust-Prinzipien	Nicht spezifisch	Nein	Zu breite Rahmung, kein Artefakt
Dakić et al. (2024)	Azure allgemein	Zero-Trust-Architektur	Nicht spezifisch	Nein	Nicht AKS-/container-spezifisch
Nogueira/Resende (2026)	Kubernetes Admission Control	Nicht adressiert	OPA/Gatekeeper, CI/CD	Nein	Kein Identitätsfokus
Bu Haimed et al. (2023)	Nicht spezifisch	Privilege Escalation (Azure AD)	Nicht adressiert	Nein	Keine Policy-Durchsetzung
Morić et al. (2025)	Kubernetes Control Plane	Zugriffskontrolle	Compliance-Assessment	Nein	Kein automatisiertes Enforcement

Wie Tabelle 2 verdeutlicht, adressiert die nächstverwandte Arbeit von Kripa zwar sowohl AKS als auch föderierte Identität und Policy-Engine-Durchsetzung, eine systematische Übersetzung von STRIDE- oder MITRE-ATT&CK-Bedrohungen in konkrete Regeln ist anhand von Titel und Themenfokus der Veröffentlichung jedoch nicht erkennbar (Vgl. Kripa, 2025). Eine seitengenaue Verifikation dieser Einordnung ist bei dieser Konferenzquelle mangels durchgehender Paginierung nicht möglich, weshalb die Tabelle die Bedrohungsmodell-Zuordnung als „nicht erkennbar“ statt als gesicherte Verneinung ausweist und die Abgrenzung der vorliegenden Arbeit nicht auf dieser Einzelquelle allein ruht. Da bereits keine der im Folgenden diskutierten vier Arbeiten AKS-Fokus, Identitätsbezug und Policy-as-Code-Umsetzung gemeinsam abdeckt, trägt dieses Vier-Quellen-Bündel die Neuheitsbegründung unabhängig von der Einzeleinschätzung zu Kripa. Die Arbeit von Kripa erschien auf einer kleineren, regionalen Studierendenkonferenz und wird deshalb als Praxisnähe-Beleg für die grundsätzliche Umsetzbarkeit eines solchen Ansatzes gewertet, nicht als alleiniger wissenschaftlicher Ankerpunkt der Abgrenzung.

Arbeiten zu Zero-Trust-Prinzipien in Kubernetes bleiben demgegenüber auf konzeptioneller Ebene, ohne ein durchsetzbares Artefakt zu entwickeln (Vgl. dos Santos, 2025, S. 57–71). Policy-zentrierte Arbeiten wie die Untersuchung einer gemeinsamen Policy-as-Code-Definitionsschicht über CI/CD und Admission Control hinweg liefern demgegenüber belastbare Evaluationsmethodik, berücksichtigen dabei aber weder Identitäten noch Token-Missbrauch (Vgl. Nogueira & Resende, 2026, Art.-Nr. 453).

Aus dieser Bestandsaufnahme ergeben sich drei Abgrenzungen, die gemeinsam, nicht primär über die Arbeit von Kripa, die Position der geplanten Masterarbeit tragen. Erstens grenzt sich die Arbeit von einem allgemeinen Zero-Trust-Ansatz für Kubernetes- oder Cloud-Umgebungen ab, der für den vorgesehenen Untersuchungsrahmen zu unspezifisch wäre (Vgl. Dakić et al., 2024, Art.-Nr. 2). Zweitens grenzt sie sich von einem rein policy-zentrierten Ansatz ohne Identitätsfokus ab, der die Eskalationsproblematik über Workload-Identitäten und Token-Missbrauch unzureichend adressiert (Vgl. Nogueira & Resende, 2026, Art.-Nr. 453). Drittens grenzt sie sich von einem Compliance-Assessment ohne automatisierte Regeldurchsetzung ab, das Zugriffskontrollen zwar bewertet, aber nicht selbst erzwingt (Vgl. Morić, Dakić & Čavala, 2025, Art.-Nr. 30).

Innerhalb dieses Vier-Quellen-Bündels, ergänzt um die als Praxisnähe-Beleg gewertete, für die Neuheitsbegründung aber nicht notwendige Arbeit von Kripa, beansprucht die geplante Masterarbeit keine grundsätzlich neue Fragestellung. Sie positioniert sich stattdessen über eine methodisch-operationale Verengung auf genau zwei vollständig konstruierte, demonstrierte und evaluierte Bedrohungen, eine reproduzierbare Übersetzungskette von Bedrohungsmodell zu Regelwerk und eine daran gekoppelte Wirksamkeitsmessung.

5 Vorläufige Gliederung der Arbeit

Die vorläufige Gliederung der Masterarbeit folgt dem am Fachbereich vorgegebenen Funktionsschema aus Einleitung, Grundlagen, Umsetzung und Schluss. Das Kapitel Einleitung nimmt rund 12,5 Prozent des Gesamtumfangs ein und führt den Problemkontext der Identitäts- und Privilegien-Eskalationsrisiken in Azure Kubernetes Service, die Forschungsfrage sowie einen Überblick über die Design-Science-Research-Methodik ein. Das Kapitel Theoretische Grundlagen und Stand der Technik umfasst etwa 25 Prozent des Umfangs und stellt die AKS-Architektur, Identitäts- und Berechtigungsmodelle, die Bedrohungsmodelle STRIDE und MITRE ATT&CK for Containers sowie Policy-as-Code-Konzepte und die vorgesehenen Benchmark- und Grundschutz-Standards dar.

Der mit rund 48,5 Prozent größte Block der Umsetzung verteilt sich auf die Kapitel Methodik und Projektumsetzung. Das Methodik-Kapitel begründet die Wahl der Design-Science-Research-Methodologie nach Hevner et al. sowie das Prozessmodell nach Peffers et al. (Vgl. Hevner et al., 2004, S. 75–105). Das Kapitel Projektumsetzung gliedert sich entlang der Design-Science-Feinstruktur in Artefaktanforderungen, Artefaktkonstruktion und Design, Demonstration sowie Evaluation. Die Artefaktanforderungen leiten sich aus der Bedrohungsanalyse und den Standards des Grundlagenkapitels ab, die Artefaktkonstruktion beschreibt die Umsetzung des Policy-as-Code-Frameworks auf Basis von Microsoft Entra Workload ID und dem OPA-Gatekeeper-basierten Azure-Policy-Add-on für AKS. Die Demonstration erfolgt anhand der auf die zwei priorisierten Bedrohungen zugeschnittenen AKS-Referenzarchitektur, die Evaluation misst das Compliance-Delta gegenüber dem CIS-Benchmark sowie die Ergebnisse der mit Stratus Red Team durchgeführten Breach & Attack Emulation für beide Bedrohungen.

Der Schlussblock nimmt etwa 14 Prozent des Umfangs ein und verteilt sich auf die Kapitel Bewertung und Diskussion sowie Fazit und Ausblick. Bewertung und Diskussion ordnet die Evaluationsergebnisse gegenüber der Forschungsfrage und den Artefaktanforderungen ein. Fazit und Ausblick fasst die Ergebnisse zusammen, ohne neue Fakten oder Quellen einzuführen, und greift Fragestellungen auf, die sich neu aus der Evaluation ergeben, etwa die Übertragbarkeit des Frameworks auf andere Kubernetes-Distributionen oder Cloud-Anbieter. Ergänzend verortet Fazit und Ausblick die aus dem Artefaktumfang ausgeschlossene dritte priorisierte Bedrohung, RBAC-Fehlkonfigurationen mit Verletzung des Least-Privilege-Prinzips, als konzeptionellen Ausblick, der skizziert, wie sich das entwickelte Policy-as-Code-Framework grundsätzlich auf diese Bedrohung übertragen ließe, ohne sie selbst zu implementieren und zu evaluieren.

Daraus ergibt sich folgende vorläufige Kapitelstruktur:

1. Einleitung
2. Theoretische Grundlagen und Stand der Technik
3. Methodik
4. Projektumsetzung
5. Bewertung und Diskussion
6. Fazit und Ausblick

6 Zeit- und Projektplanung

Die Bearbeitung der Masterarbeit beginnt am 1. November 2026 und endet mit der Abgabe am 1. April 2027, was einer Bearbeitungsdauer von rund fünf Monaten entspricht. Die Zeitplanung orientiert sich an den sechs Phasen der Design Science Research Methodology nach Peffers et al. (Vgl. Peffers et al., 2007, S. 45–77), wodurch Methodik und Projektplanung durchgängig dieselbe Terminologie verwenden. Tabelle 3 ordnet den DSRM-Phasen konkrete Zeiträume und Meilensteine zu und weist zusätzlich einen expliziten Pufferzeitraum innerhalb der Design-and-Development-Phase aus.

Tabelle 3: Zeit- und Projektplanung entlang der DSRM-Phasen mit explizitem Pufferzeitraum

DSRM-Phase	Zeitraum	Meilenstein
Problem Identification & Objectives	01.11.–30.11.2026	Bedrohungsanalyse (STRIDE, MITRE ATT&CK), Anforderungsableitung
Design and Development	01.12.2026–14.02.2027	Konstruktion des auf zwei priorisierte Bedrohungen begrenzten Policy-as-Code-Frameworks, Aufbau der minimalen AKS-Referenzarchitektur
Pufferzeitraum	15.02.–21.02.2027	Reservezeit für Verzögerungen bei der Einarbeitung in Rego, Open Policy Agent oder Microsoft Entra Workload ID, kein eigener Liefergegenstand
Demonstration	22.02.–07.03.2027	Lab-Proof-of-Concept mit Angriffsszenarien (Token-Diebstahl, Privilege Escalation über Managed Identities), parallele Rohfassung des Demonstrations-Unterkapitels
Evaluation	08.03.–28.03.2027	Compliance-Delta-Messung, Auswertung der Breach & Attack Emulation, parallele Rohfassung der Kapitel Evaluation, Bewertung und Diskussion sowie Fazit anhand vorliegender Teilergebnisse
Communication	29.03.–01.04.2027	Überarbeitung der Rohfassungen zur Endfassung, Korrektur, Abgabe am 1. April 2027

Die Design-and-Development-Phase umfasst rund elf Wochen für die Konstruktion des Artefakts und den Aufbau der AKS-Referenzarchitektur, ergänzt um einen eigens ausgewiesenen, einwöchigen Pufferzeitraum vom 15. bis 21. Februar 2027 unmittelbar vor Beginn der Demonstration. Der Pufferzeitraum ist für Verzögerungen reserviert, die aus der Einarbeitung in Rego, Open Policy Agent oder Microsoft Entra Workload ID entstehen können, definiert dafür aber keinen eigenen Liefergegenstand. Verläuft die Konstruktion planmäßig, steht die dadurch freibleibende Zeit für eine frühere Fertigstellung der parallel begonnenen Rohfassungen zur Verfügung. Das Zeitbudget für Konstruktion und Pufferzeitraum speist sich aus der Demonstrations- und Evaluationsphase, die durch die Begrenzung auf zwei priorisierte Bedrohungen einen vollständigen Simulationsdurchlauf sowie eine Compliance-Delta-Messung und eine Angriffsauswertung weniger umfasst als bei einer breiteren Bedrohungsauswahl. Die Verschriftlichung der Kapitel Einleitung und Theoretische Grundlagen und Stand der Technik beginnt bereits parallel zur Bedrohungsanalyse und zum technischen Aufbau, nicht erst in der Communication-Phase.

Für die evaluationsabhängigen Kapitel Projektumsetzung, Bewertung und Diskussion sowie Fazit und Ausblick gilt eine gestaffelte Rohfassung-vor-Endfassung-Logik statt einer Verschriftlichung erst nach Abschluss der Evaluation. Die Unterkapitel Artefaktanforderungen und Artefaktkonstruktion und Design entstehen bereits während der Design-and-Development-Phase, das Unterkapitel Demonstration wird parallel zum Lab-Proof-of-Concept im Februar und Anfang März 2027 auf Basis der dort laufend protokollierten Zwischenergebnisse in einer ersten Fassung verschriftlicht. Für Evaluation, Bewertung und Diskussion sowie Fazit und Ausblick wird bereits während der Evaluationsphase im März 2027 eine vorläufige Rohfassung angelegt, die auf den zu diesem Zeitpunkt bereits vorliegenden Teilergebnissen des Compliance-Delta und der zuerst abgeschlossenen Angriffssimulation aufbaut.

Erst nach Abschluss beider Angriffssimulationen und der vollständigen Compliance-Delta-Messung Ende März 2027 werden diese Rohfassungen zu einer Endfassung überarbeitet. Für die Communication-Phase verbleibt dadurch nicht die vollständige Neuverschriftlichung der evaluationsabhängigen Kapitel, sondern deren gezielte Überarbeitung anhand der finalen Ergebnisse sowie ein abschließender Korrekturdurchgang. Diese gestaffelte Vorgehensweise reduziert die zeitliche Konzentration der Schreibarbeit auf die letzten Tage vor der Abgabe, ohne den Gesamtzeitrahmen von fünf Monaten zu verändern.

Die AKS-Testumgebung wird über das Studierendenkontingent Azure for Students finanziert, wodurch der Kostenrahmen des Proof-of-Concept auf einen niedrigen dreistelligen Betrag begrenzt bleibt. Da Microsoft Defender for Containers und Defender for Cloud Security Posture Management laufend abgerechnete Dienste sind, wird die Referenzumge-

bung nach Abschluss der jeweiligen Testphase pausiert beziehungsweise abgebaut, statt durchgehend über Demonstration und Evaluation hinweg aktiv zu bleiben. Der reproduzierbare Auf- und Abbau erfolgt über dieselben Bicep-Vorlagen, die bereits die minimale AKS-Referenzarchitektur beschreiben, wodurch sich die Pausierung ohne manuellen Konfigurationsaufwand wiederholt durchführen lässt. Am Ende des Pufferzeitraums, unmittelbar vor Beginn der Demonstration, erfolgt zusätzlich ein Budget-Checkpoint, der den bis dahin verbrauchten Anteil des Guthabens gegen den verbleibenden Bedarf für Demonstration und Evaluation prüft und bei Bedarf eine Anpassung des Laborbetriebs ermöglicht. Die Zuordnung von Erst- und Zweitgutachter steht zum jetzigen Zeitpunkt noch aus und wird vor Bearbeitungsbeginn geklärt.

Literaturverzeichnis

Ali, A., Imran, M., Kuznetsov, V., Trigazis, S., Pervaiz, A., Pfeiffer, A., & Mascheroni, M. (2024). Implementation of New Security Features in CMSWEB Kubernetes Cluster at CERN. *26th International Conference on Computing in High Energy & Nuclear Physics (CHEP 2023), EPJ Web of Conferences, Vol. 295*, 07026. https://doi.org/10.1051/epjconf/202429507026

Bu Haimed, I., Albahar, M., & Alzubaidi, A. (2023). Exploiting Misconfiguration Vulnerabilities in Microsoft's Azure Active Directory for Privilege Escalation Attacks. *Future Internet*, *15*(7), 226. https://doi.org/10.3390/fi15070226

Dakić, V., Morić, Z., Kapulica, A., & Regvart, D. (2024). Analysis of Azure Zero Trust Architecture Implementation for Mid-Size Organizations. *Journal of Cybersecurity and Privacy*, *5*(1), 2. https://doi.org/10.3390/jcp5010002

dos Santos, R. F. G. (2025). Applying Zero Trust to Kubernetes Clusters. *ARIS2-Journal (Advanced Research on Information Systems Security)*, *5*(1), 57–71. https://doi.org/10.56394/aris2.v5i1.58

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Kripa, R. K. (2025). SecureCloudOps: Zero-Trust Identity Federation and Runtime Monitoring in Azure Kubernetes Service (AKS). *World Conference on Cutting-Edge Science & Technology (WCCEST 2025)*. https://doi.org/10.1109/WCCEST66994.2025.11390007

Lee, S., & Lee, J. (2022). Kubernetes of Cloud Computing Based on STRIDE Threat Modeling. *Journal of the Korea Institute of Information and Communication Engineering*, *26*(7), 1047–1059. https://doi.org/10.6109/jkiice.2022.26.7.1047

March, S. T., & Smith, G. F. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems*, *15*(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Morić, Z., Dakić, V., & Čavala, T. (2025). Security Hardening and Compliance Assessment of Kubernetes Control Plane and Workloads. *Journal of Cybersecurity and Privacy*, *5*(2), 30. https://doi.org/10.3390/jcp5020030

Nogueira, L., & Resende, A. (2026). Reusing Policy-as-Code Across CI/CD and Kubernetes Admission Control: An Empirical Assessment of Governance Consistency. *Computers*, *15*(7), 453. https://doi.org/10.3390/computers15070453

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/mis0742-1222240302

Souppaya, M., Morello, J., & Scarfone, K. (2017, 25. September). *Application Container Security Guide*. National Institute of Standards und Technology (NIST). https://doi.org/10.6028/nist.sp.800-190

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. https://doi.org/10.1057/ejis.2014.36

Internetquellen

Bundesamt für Sicherheit in der Informationstechnik (BSI). (2023). *APP.4.4 Kubernetes (IT-Grundschutz-Kompendium, Edition 2023)*. Verfügbar 2026-08-22 unter https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/IT-GS-Kompendium_Einzel_PDFs_2023/06_APP_Anwendungen/APP_4_4_Kubernetes_Edition_2023.html

Center for Internet Security (CIS). (2026, Juni). *CIS Microsoft Azure Kubernetes Service (AKS) Benchmark*. Verfügbar 2026-08-22 unter https://www.cisecurity.org/benchmark/kubernetes

MITRE Center for Threat-Informed Defense. (2021, 3. Mai). *ATT&CK for Containers*. Verfügbar 2026-08-21 unter https://ctid.mitre.org/projects/attack-for-containers/

Shamim, M. S. I., Bhuiyan, F. A., & Rahman, A. (2020, 27. Juni). *XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices*. arXiv: 2006.15275 [cs.CR].

Weizman, Y. (2020, 2. April). *Threat Matrix for Kubernetes*. Verfügbar 2026-08-21 unter https://www.microsoft.com/en-us/security/blog/2020/04/02/attack-matrix-kubernetes/

Wong, A. Y., Chekole, E. G., Ochoa, M., & Zhou, J. (2021, 22. November). *Threat Modeling and Security Analysis of Containers: A Survey*. arXiv: 2111.11475 [cs.CR].

KI-Hilfsmittelverzeichnis

Tool	Beschreibung	Einsatzbereich	Version
[KI-Tool eintragen]	[Beschreibung eintragen]	[Einsatzbereich eintragen]	[Version eintragen]

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die angemeldete Prüfungsleistung in allen Teilen eigenständig ohne Hilfe von Dritten anfertigen und keine anderen als die in der Prüfungsleistung angegebenen Quellen und zugelassenen Hilfsmittel verwenden werde. Sämtliche wörtlichen und sinngemäßen Übernahmen inklusive KI-generierter Inhalte werde ich kenntlich machen. Diese Prüfungsleistung hat zum Zeitpunkt der Abgabe weder in gleicher noch in ähnlicher Form, auch nicht auszugsweise, bereits einer Prüfungsbehörde zur Prüfung vorgelegen; hiervon ausgenommen sind Prüfungsleistungen, für die in der Modulbeschreibung ausdrücklich andere Regelungen festgelegt sind. Mir ist bekannt, dass die Zuwiderhandlung gegen den Inhalt dieser Erklärung einen Täuschungsversuch darstellt, der das Nichtbestehen der Prüfung zur Folge hat und daneben strafrechtlich gem. § 156 StGB verfolgt werden kann. Darüber hinaus ist mir bekannt, dass ich bei schwerwiegender Täuschung exmatrikuliert und mit einer Geldbuße bis zu 50.000 EUR nach der für mich gültigen Rahmenprüfungsordnung belegt werden kann. Ich erkläre mich damit einverstanden, dass diese Prüfungsleistung zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

, 23.8.2026

(Ort, Datum)

Philipp Rose

(Eigenhändige Unterschrift)