oneconsult
together against cyberattacks

Agentische KI-Sicherheit

Bedrohungen, Angriffsvektoren und praxisnahe Absicherungsstrategien für KI-Applikationen

Whitepaper von

Dr. Marco Bertenghi

April 2026

Abstract

Die rasante Verbreitung künstlicher Intelligenz (KI) beeinflusst Geschäftsprozesse grundlegend. KI-Agenten handeln zunehmend autonom, rufen APIs auf, nutzen interne Tools und treffen eigenständige Entscheidungen. Diese Autonomie bietet enormes Potenzial, eröffnet gleichzeitig aber auch eine neuartige und weitläufige Angriffsfläche für Cyberkriminelle, die mit herkömmlichen Sicherheitstests nicht abgedeckt werden kann.

Dieses Whitepaper bietet einen umfassenden Überblick über die Bedrohungslandschaft agentischer KI-Systeme. Es definiert zentrale Begriffe, analysiert die wichtigsten Angriffsvektoren und ordnet diese in aktuelle Sicherheitsstandards wie die OWASP-Sicherheitskataloge für KI und agentische Systeme ein. Darüber hinaus werden praxisnahe und realistische Absicherungsstrategien vorgestellt.

Zentrale Erkenntnisse

Prompt-Injektionen bleiben der gefährlichste Angriffsvektor für KI-Applikationen und werden durch agentische Systeme massiv verstärkt.

Agentische Frameworks mit Tool-Zugriff, Planungsfähigkeit und Memory schaffen eine völlig neue Kategorie von Sicherheitsrisiken, von Agent Goal Hijacking bis hin zu Memory Poisoning.

Herkömmliche Sicherheitstests reichen nicht aus. Neuartige Schwachstellen erfordern spezialisierte Penetration Tests mit Fokus auf KI-Sicherheit.

Defense in Depth ist der Schlüssel, denn einzelne Schutzmechanismen genügen nicht.
Nur mehrschichtige Ansätze kombiniert mit dem Prinzip der minimalen Berechtigungen bieten den besten Schutz.

Proaktives Handeln ist entscheidend.
Organisationen müssen ihre KI-Applikationen testen und absichern, bevor Angreifer Schwachstellen aktiv ausnutzen können.

Dr. Marco Bertenghi

Senior Penetration Tester

Ausbildung

Dr. Marco Bertenghi promovierte an der Zurich Graduate School in Mathematics (ZGSM, UZH/ETHZ) im Fachbereich Mathematik mit einer Spezialisierung auf Gedächtnis- und Diffusionsprozesse, inspiriert durch die Dynamik in Systemen der künstlichen Intelligenz (KI). Diese solide analytische und mathematische Grundlage hat er seither nahtlos in den Bereich der Cybersicherheit überführt.

Praktische Erfahrung

Seit 2023 ist Marco Bertenghi als Penetration Tester (seit 2026 Senior) bei der Oneconsult AG tätig, wo er mit Sicherheitstests aktiv zur Stärkung der Cyberresilienz von Unternehmen beiträgt. Er hat bereits mehrere Projekte mit KI-Sicherheitsfokus geleitet – unter anderem in der Versicherungsbranche, im Bildungssektor sowie im Bankenwesen.

Zertifizierungen

Marco Bertenghi ist zertifizierter OSSTMM Professional Security Tester (OPST) sowie Burp Suite Certified Practitioner (BSCP) – Zertifizierungen, die seine akademische Ausbildung und praktische Erfahrung gezielt ergänzen.

Engagement

Darüber hinaus engagiert sich Marco Bertenghi im internen Virtual Competence Center (VCC) für den Einsatz von KI bei Oneconsult und lehrt als Gastdozent an der Fachhochschule Nordwestschweiz (FHNW) zu Penetration Testing und Red Teaming mit Schwerpunkt auf KI.

> Mit spezialisierter KI-Expertise im Bereich agentischer Systeme und LLMs sowie umfangreicher Erfahrung in der Cybersicherheit ist Marco Bertenghi ausgewiesener Experte für agentische KI-Sicherheit.

oneconsult
together against cyberattacks

Inhalt

05 Bedrohungslage für KI-Applikationen

06 - 07 Grundlagen agentischer KI-Systeme

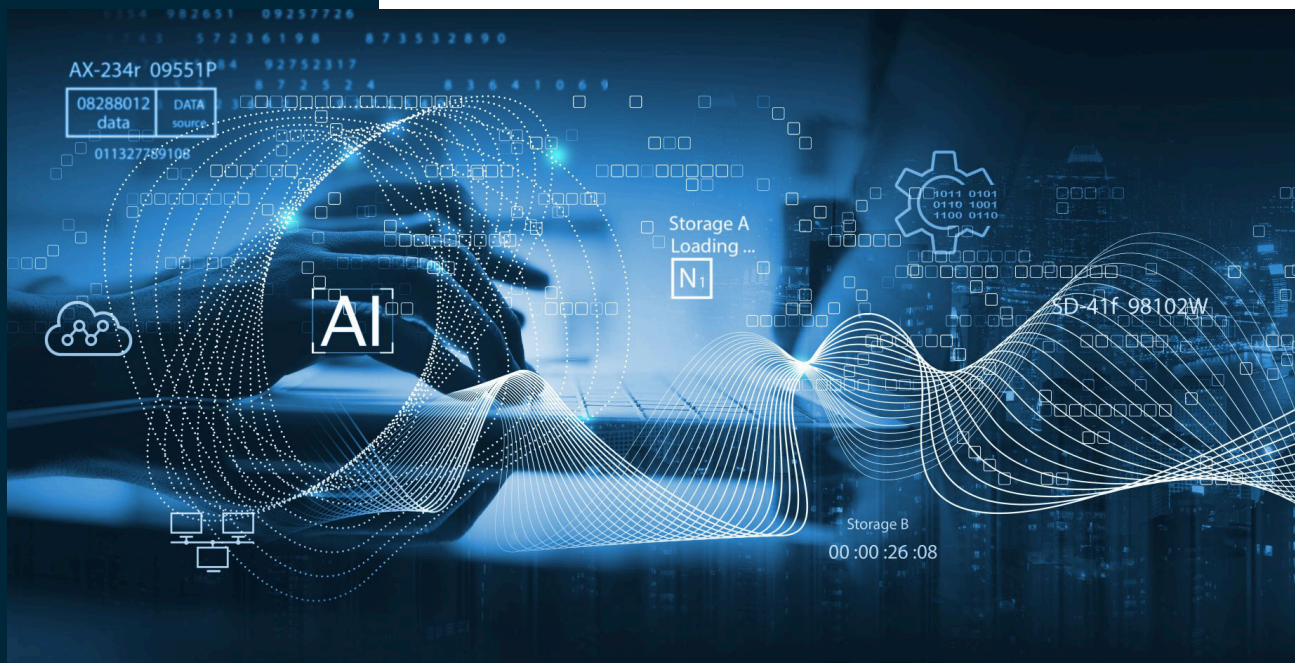
08 - 10 Angriffsvektoren im Detail

11 - 13 Praxisnahe Absicherungsstrategien

14 Grenzen klassischer Sicherheitstests

15 - 16 Fazit und Handlungsempfehlungen

17 - 20 Anhang (Glossar, Quellen, Ressourcen)



Die wachsende Bedrohungslage für KI-Applikationen

Künstliche Intelligenz (KI) hat in den vergangenen Jahren eine beispiellose Entwicklung durchlaufen. Was 2023 noch hauptsächlich aus einfachen Chatbots bestand – mit dem Paradebeispiel ChatGPT von OpenAI – hat sich zu einem Ökosystem hochkomplexer, autonomer Systeme entwickelt. Unternehmen setzen heute KI-Applikationen in diversen Geschäftsbereichen ein: von intelligenten Kundensupport-Chatbots über Retrieval-Augmented-Generation(RAG)-Systeme bis hin zu vollständig autonomen KI-Agenten, die eigenständig Aufgaben planen, Tools aufrufen und Entscheidungen treffen.

Mit dieser zunehmenden Verbreitung steigt die Angriffsfläche dramatisch. Die OWASP LLM Top 10[1] – ein international anerkannter Katalog der kritischsten Sicherheitsrisiken für KI-Sprachmodelle – stufen gezielte Manipulationen durch eingeschleuste Anweisungen als das grösste Risiko ein. Doch agentische KI-Systeme potenzieren diese Bedrohung um ein Vielfaches: Ein erfolgreich manipulierter Agent kann nicht nur Informationen preisgeben, sondern aktiv schädliche Aktionen ausführen, auf interne Systeme zugreifen und seine Privilegien missbrauchen.

Die Realität zeigt, dass diese Risiken keine theoretischen Szenarien sind. Im Januar 2026 wurden für den KI-Agenten OpenClaw innerhalb von 72 Stunden nach der Veröffentlichung exponierte Admin-Panels, gestohlene API-Keys und aktive Infostealer-Kampagnen entdeckt[2]. Bereits zuvor demonstrierten Sicherheitsforscher, dass populäre Coding-Agenten wie GitHub Copilot und Google Jules durch Prompt-Injektionen vollständig kompromittiert werden konnten, einschliesslich der Ausführung von Schadcode.

Agentische KI-Systeme handeln autonom, mit weitreichenden Zugriffsrechten und oft tief in der Unternehmensinfrastruktur. **Ein kompromittierter Agent ist somit kein passives Datenleck, sondern ein aktiver Angreifer innerhalb Ihres Netzwerks.**

Dieses Whitepaper richtet sich an Entscheidungsträger, Sicherheitsverantwortliche und technische Führungskräfte, die KI-Applikationen in ihren Organisationen einsetzen oder deren Einsatz planen.

Es verfolgt drei zentrale Ziele:

1. **Sensibilisierung:** Ein fundiertes Verständnis der Bedrohungslandschaft agentischer KI-Systeme vermitteln, gestützt auf aktuelle Forschungsergebnisse und reale Sicherheitsvorfälle.
2. **Orientierung:** Die wichtigsten Sicherheitsstandards und Bewertungsrahmen vorstellen, darunter die OWASP LLM Top 10[3], die OWASP Top 10 for Agentic Applications[4] und die OWASP MCP Top 10[5].
3. **Handlungsfähigkeit:** Konkrete, praxisnahe und realistische Absicherungsstrategien aufzeigen, die Organisationen sofort umsetzen können.

Nach der Lektüre dieses Whitepapers werden Sie in der Lage sein, die Sicherheitsrisiken Ihrer KI-Applikationen realistisch einzuschätzen, geeignete Schutzmassnahmen zu priorisieren und fundierte Entscheidungen über spezialisierte Sicherheitstests zu treffen.

Grundlagen agentischer KI-Systeme

Um die Sicherheitsrisiken agentischer KI-Systeme zu verstehen, ist eine klare Abgrenzung der verschiedenen Systemtypen und ihrer Architekturkomponenten notwendig.

Von Chatbots zu autonomen Agenten

KI-Applikationen lassen sich nach ihrem Autonomiegrad in drei Kategorien einteilen:

Chat-Applikationen (LLMs)

Der Benutzer interagiert direkt mit einem Sprachmodell, wie zum Beispiel ChatGPT.

Die Anwendung verarbeitet Texteingaben und gibt Textantworten zurück. In einigen Fällen gibt es mehre Modalitäten, wie zum Beispiel Bild, Audio und Video. Sicherheitsrisiken betreffen primär Prompt-Injektionen und die Offenlegung sensibler Daten.

RAG-Systeme (Retrieval-Augmented Generation)

Das Sprachmodell wird um externe Datenquellen ergänzt, wie zum Beispiel bei Google NotebookLM.

Bei jeder Anfrage werden relevante Dokumente aus einer Wissensbasis abgerufen und als Kontext mitgegeben. Zusätzlich zu den LLM-Risiken entstehen Angriffsvektoren durch manipulierte Dokumente und unsichere Daten-Pipelines.

Agentische Systeme

KI-Agenten, wie zum Beispiel aw oder GPT-Codex, handeln autonom.

Sie planen mehrstufige Aufgaben, rufen externe Tools und APIs auf, speichern Informationen im Gedächtnis und delegieren Aufgaben an andere Agenten. Diese Systeme besitzen die grösste Angriffsfläche, da erfolgreiche Angriffe zu unkontrollierten Aktionen mit realen Auswirkungen führen können.

Die Autonomie agentischer Systeme ist keine optionale Eigenschaft, sondern ihre zentrale Daseinsberechtigung. Abgesehen von einfachen Chatbot-Anwendungen bieten LLMs allein Unternehmen nur einen begrenzten Mehrwert. Erst die Fähigkeit, eigenständig zu planen, Entscheidungen zu treffen und in Systeme einzugreifen, entfaltet ihr volles wirtschaftliches Potenzial.

Genau diese Autonomie versetzt agentische Systeme jedoch in eine hochprivilegierte Position innerhalb der IT-Landschaft und macht sie damit zu einem äusserst attraktiven Ziel für Angreifer.

Architekturkomponenten agentischer Systeme

Ein typisches agentisches KI-System besteht aus mehreren Kernkomponenten, die jeweils eigene Sicherheitsrisiken mit sich bringen:

Komponente	Funktion	Sicherheitsrisiken
LLM (Denkmaschine)	Verarbeitung natürlicher Sprache, Entscheidungsfindung	*Prompt-Injektionen, Jailbreaks*
LLM (Planungsmodul)	Zerlegung komplexer Aufgaben in Teilschritte	*Goal Hijacking, Plan Manipulation*
Tool-Integration	Zugriff auf APIs, Datenbanken, Dateisysteme	*Tool Misuse, Rug Pull, Privilege Escalation*
Memory/Kontext	Speicherung von Informationen über Sitzungen hinweg	*Memory Poisoning, Context Leakage und Context Rot*
Multi-Agent-Orchestrierung	Delegation und Kommunikation zwischen Agenten	*Agent Communication Poisoning, Rogue Agents*

Die Bedrohungslandschaft: Angriffsvektoren im Detail

Die Bedrohungslandschaft für KI-Applikationen ist vielschichtig. Im Folgenden werden die wichtigsten Angriffsvektoren systematisch dargestellt, beginnend mit den am häufigsten ausgenutzten bis zu den technisch anspruchsvollsten.

Prompt-Injektionen: der gefährlichste Angriffsvektor

Prompt-Injektionen stehen an erster Stelle der OWASP LLM Top 10 und gelten weiterhin als der meistgenutzte Angriffsvektor gegen KI-Applikationen. Dabei manipulieren Angreifer die Eingaben an ein KI-System, um Sicherheitsmechanismen zu umgehen, das vorgesehene Verhalten zu verändern oder sensible Daten abzugreifen.

Es werden drei Hauptkategorien unterschieden:

Direkte Prompt-Injektionen

Der Angreifer gibt bösartige Anweisungen direkt in das Benutzereingabefeld ein, um System-Prompts offenzulegen, Sicherheitsregeln zu umgehen oder unautorisierte Aktionen auszulösen. **Trotz zahlreicher Schutzmechanismen zeigen aktuelle Studien, dass selbst fortgeschrittene Verteidigungen mit einer Erfolgsrate von über 70 Prozent umgangen werden können.**

Indirekte Prompt-Injektionen

Schadhafte Anweisungen werden in Daten eingebettet, die das KI-System aus externen Quellen verarbeitet, etwa in Webseiten, Dokumenten, E-Mails oder Datenbankeinträgen. Dies ist besonders kritisch für RAG-Systeme und Agenten mit Internetzugang, da Angreifer keinen direkten Zugang zum System benötigen.

Multi-Turn-Angriffe

Der Angreifer verteilt die bösartige Anweisung über mehrere aufeinanderfolgende Nachrichten, um kontextbasierte Schutzmechanismen zu umgehen. Jede einzelne Nachricht erscheint harmlos, in der Gesamtheit führen sie jedoch zur gewünschten Manipulation.

Aktuelle Forschung zeigt: Die meisten publizierten Verteidigungsmechanismen bieten ein falsches Sicherheitsgefühl. In der Studie "The Attacker Moves Second"[6] wurden alle 12 untersuchten Verteidigungen mit einer Erfolgsrate von über 70 Prozent, teils sogar über 90 Prozent, umgangen.

Angriffe auf agentische Frameworks

Agentische KI-Systeme erweitern die Angriffsfläche erheblich.

Die OWASP Top 10 for Agentic Applications identifizieren unter anderem folgende Angriffsvektoren:

Agent Goal Hijacking (Intent Hijacking):

Durch manipulierte Eingaben wird der Agent dazu gebracht, ein völlig anderes Ziel zu verfolgen als von den Entwicklern beabsichtigt. Bei einem Agenten mit Tool-Zugriff kann dies dazu führen, dass er bösartige Aktionen ausführt, etwa Dateien löscht, Daten exfiltriert oder schädlichen Code ausführt.

Tool Misuse und Privilege Escalation:

Agenten mit Zugriff auf Tools und APIs können manipuliert werden, um diese über den vorgesehenen Rahmen hinaus zu nutzen. Insbesondere bei unzureichender Berechtigungskontrolle kann ein Agent privilegierte Aktionen ausführen, für die er nicht autorisiert ist.

Memory Poisoning:

Angreifer manipulieren den persistenten Speicher eines Agenten, um dessen zukünftiges Verhalten zu beeinflussen. Angriffe wie SpAIware[7] und ZombieAgent[8] demonstrieren, dass einmal eingeschleuster Schadcode über Sitzungsgrenzen hinweg persistent bleibt und bei späteren Interaktionen aktiviert wird.

Agent Communication Poisoning:

In Multi-Agent-Systemen können Angreifer die Kommunikation zwischen Agenten manipulieren. Forschungsergebnisse zeigen, dass alle der untersuchten LLMs anfällig für derartige Inter-Agent-Kompromittierungen sind[9].

Overwhelming Human-in-the-Loop:

Agentische Systeme können den menschlichen Überwachungsprozess überlasten, indem sie Benutzer mit übermässig vielen Bestätigungsanfragen konfrontieren. Dies führt zu Entscheidungsmüdigkeit und letztlich zur blinden Genehmigung potenziell schädlicher Aktionen.

MCP-Protokoll: neue Angriffsfläche durch Standardisierung

Das Model Context Protocol (MCP)[10] von Anthropic hat sich als De-facto-Standard für die Tool-Integration in KI-Systemen etabliert. Die zunehmende Verbreitung bringt jedoch spezifische Sicherheitsrisiken mit sich, die in der OWASP MCP Top 10 dokumentiert sind:

Tool Poisoning:

Ein bösartiger MCP-Server liefert manipulierte Tool-Beschreibungen oder Tool-Ergebnisse, die versteckte Anweisungen enthalten. Der Agent führt diese Anweisungen aus, ohne dass Benutzer es bemerken, da die schadhaften Instruktionen in den Metadaten verborgen sind.

Rug-Pull-Angriffe:

Ein zunächst vertrauenswürdiger MCP-Server ändert nach der Installation sein Verhalten und führt bösartige Aktionen aus. Da MCP-Server remote gehostet werden können, kann der Serverbetreiber das Verhalten jederzeit ändern.

Transitive Vertrauensangriffe:

Vertrauen, das einem MCP-Server gewährt wird, wird implizit auf andere, potenziell nicht vertrauenswürdige Server übertragen. Dies ermöglicht Angriffsketten über mehrere Server hinweg.

RAG-Pipeline und Output-Handling

Retrieval-Augmented Generation (RAG) erweitert KI-Systeme um externe Datenquellen und schafft dabei neue Angriffsvektoren:

Document Poisoning:

Angreifer schleusen manipulierte Dokumente in die Wissensbasis ein. Der PoisonedRAG-Angriff[11] erreicht bei fünf vergifteten Dokumenten eine Erfolgsrate von über 90 Prozent, indem der Kontext gezielt manipuliert wird. Auch das US-amerikanische Unternehmen Anthropic mit Fokus auf KI hat in diesem Bereich geforscht und gezeigt, dass bereits wenige vergiftete Beispiele ausreichen, um das Verhalten eines Modells gezielt zu beeinflussen[12].

Unsicheres Output-Handling:

Die Ausgaben des KI-Systems werden von nachgelagerten Komponenten wie Renderern, Browsern oder Shells weiterverarbeitet. Manipulierte Ausgaben können klassische Angriffsvektoren wie Cross-Site Scripting (XSS), Server-Side Request Forgery (SSRF) oder Remote Code Execution (RCE) auslösen.

Datenabfluss über den Kontext:

Sensible Daten aus der Wissensbasis können durch gezielte Prompt-Manipulation exponiert werden, insbesondere wenn die Zugriffskontrolle auf Dokumentebene fehlt.

Praxisnahe Absicherungsstrategien

Die Absicherung agentischer KI-Systeme erfordert einen mehrschichtigen Ansatz. Ein einzelner Schutzmechanismus kann nicht alle Angriffsvektoren abdecken. Basierend auf aktuellen Forschungsergebnissen und Praxiserfahrungen empfehlen wir ein dreistufiges Massnahmenmodell, das nach Umsetzungsgeschwindigkeit und Wirksamkeit priorisiert ist.

Defense in Depth ist der Schlüssel. Verlassen Sie sich nie auf einen einzelnen Schutzmechanismus. Kombinieren Sie stattdessen architektonische Massnahmen mit operativen Kontrollen und kontinuierlicher Überwachung.

Woche 1-4

Sofortmassnahmen: Folgende Massnahmen bieten bei vergleichsweise geringem Aufwand den höchsten Sicherheitsgewinn und sollten priorisiert umgesetzt werden:

Prinzip der minimalen Berechtigungen (Least Privilege)

- Beschränken Sie die Zugriffsrechte jedes Agenten auf das absolute Minimum, das für seine Aufgabe erforderlich ist.
  - › Orientieren Sie sich dabei zunächst an den Berechtigungen des Auftraggebers. Einem Agenten sollten grundsätzlich nie mehr Rechte gewährt werden als den Benutzern, die ihn beauftragen. Dies dient als Obergrenze, von der aus die Rechte des Agenten weiter einzuschränken sind.
  - › Achten Sie besonders darauf, dass auch privilegierte Benutzer, wie zum Beispiel Administratoren, keine automatisch erweiterten Agentenrechte erhalten. Die aufgabenbezogene Einschränkung sollte unabhängig vom Status des Auftraggebers gelten.
- Implementieren Sie granulare Berechtigungskonzepte für jeden einzelnen Tool-Zugriff.
- Verwenden Sie kurzlebige, aufgabenspezifische Tokens anstelle von langlebigen API-Schlüsseln.

Eingabevalidierung und -bereinigung

- Implementieren Sie strikte Eingabevalidierung auf allen Ebenen: Benutzer-Prompts, System-Prompts, Tool-Eingaben und Tool-Ausgaben.
- Behandeln Sie alle Eingaben als potenziell bösartig, unabhängig von ihrer Quelle.
- Setzen Sie realistische Längenbeschränkungen für Input-Tokens.

Sichere Output-Verarbeitung

- Behandeln Sie alle LLM-Ausgaben als nicht vertrauenswürdig.
- Implementieren Sie Output-Encoding und -Sanitisierung, bevor Ergebnisse an nachgelagerte Systeme weitergegeben werden.
- Validieren Sie strukturierte Ausgaben gegen definierte Schemata. Lehnen Sie standardmässig alle Anfragen ab, die nicht dem erwarteten Schema entsprechen.
- Definieren Sie realistische Längenbeschränkungen für Output-Tokens sowie Ratenlimitierungen, insbesondere für Tool-Aufrufe. Zu grosszügig gesetzte oder fehlende Limitierungen ermöglichen nicht nur Ressourcenmissbrauch, sondern können durch exzessiv lange Ausgaben zu einem Denial-of-Wallet(DoW)-Angriff führen.

Human-in-the-Loop für kritische Aktionen

- Erzwingen Sie eine menschliche Bestätigung (HITL) für alle sicherheitskritischen Aktionen wie Datenbankschreibzugriffe, Dateiänderungen und externe API-Aufrufe.
- Gestalten Sie die Bestätigungsanfragen so, dass Entscheidungsmüdigkeit vermieden wird.

Monat 2-3

Mittelfristige Massnahmen: Diese Massnahmen erfordern mehr Planungsaufwand, bieten aber einen nachhaltigen Sicherheitsgewinn:

Sandboxing und Isolation

- Führen Sie alle Tool-Aufrufe in isolierten Umgebungen aus. Verwenden Sie Container, virtuelle Maschinen oder Mikro-VMs, um die Auswirkungen kompromittierter Agenten einzugrenzen.
- Implementieren Sie Netzwerksegmentierung, um laterale Bewegungen zu verhindern.

Umfassendes Logging und Monitoring

- Protokollieren Sie alle agentischen Aktionen lückenlos, einschliesslich Planungsschritte, Tool-Aufrufe, Inter-Agent-Nachrichten und Entscheidungspfade. Stellen Sie dabei sicher, dass Logs keine personenbezogenen Daten (Personally Identifiable Information; PII) oder sonstige sensible Informationen enthalten – etwa durch automatisierte Maskierung oder Filterung vor der Speicherung. Da eine vollständige Vermeidung technisch nicht möglich ist, sind entsprechende Schutz- und Zugriffskontrollen auf die Logdaten anzuwenden.
- Implementieren Sie Echtzeitbenachrichtigungen für anomale Verhaltensmuster.

Richtliniendurchsetzung zentralisieren

- Setzen Sie eine dedizierte Richtlinien- und Gateway-Schicht ein, die Authentifizierung, Autorisierung, Einwilligungsmanagement, Tool-Filterung und Audit-Logging konsequent durchsetzt. Durch die zentrale Auswertung aller Anfragen stellen Sie sicher, dass Ihre Sicherheitsrichtlinien über sämtliche Agenten, Server und vorgelagerte Dienste hinweg einheitlich und lückenlos greifen.

MCP-Server-Härtung

- Pinnen Sie MCP-Server-Versionen, validieren Sie Tool-Beschreibungen gegen bekannte Muster und implementieren Sie Server-Integritätsprüfungen.
- Verwenden Sie nur vertrauenswürdige, auditierte MCP-Server über verschlüsselte Kanäle (TLS 1.2+).

Spezialisierte Sicherheitstests

- Führen Sie regelmässige KI Penetration Tests durch, die speziell auf die neuartigen Angriffsvektoren agentischer Systeme ausgerichtet sind. **Herkömmliche Schwachstellen-Scans können diese Risiken nicht zuverlässig abdecken.**

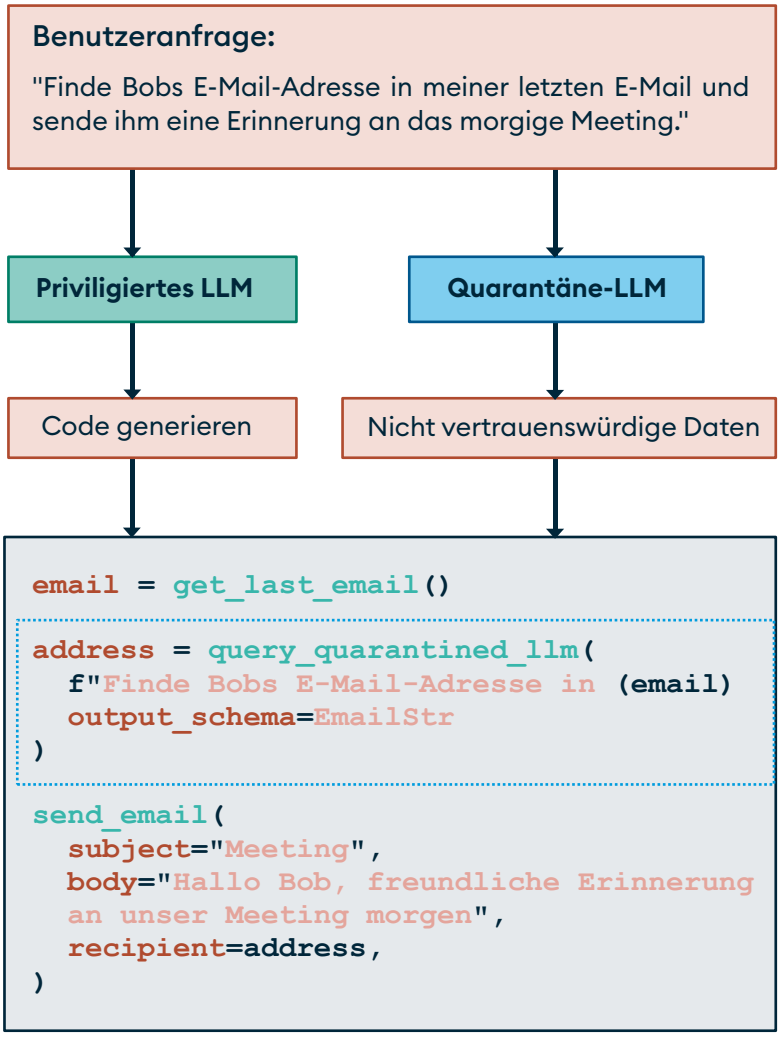
Ab Quartal 2

Strategische Massnahmen: Diese Massnahmen erfordern architektonische Veränderungen sowie langfristige Investitionen und müssen daher strategisch geplant und umgesetzt werden.

Dual-LLM-Architektur

Trennen Sie das Sprachmodell, das Benutzeranfragen verarbeitet, vom Modell, das sicherheitskritische Entscheidungen trifft. Ansätze wie CaMeL (**C**apabilities for **M**achin**e** **L**earning) von Google DeepMind[13] nutzen ein separates, nicht durch Benutzereingaben beeinflussbares Modell für die Autorisierung von Tool-Aufrufen.

Beispiel einer dualen LLM-Architektur eingeführt von Simon Willison[14] und verwendet für den CaMeL-Ansatz



Die duale Architektur besteht aus einem privilegierten LLM und einem Quarantäne-LLM:

- Das *privilegierte LLM* erhält Instruktionen, plant Aktionen und kann Tools ausführen.
- Das *unter Quarantäne gestellte LLM* kann vom privilegierten LLM aufgerufen werden, sobald nicht vertrauenswürdige Daten prozessiert werden müssen. Das Quarantäne-LLM kann keine Tools ausführen, es generiert lediglich Text.

Im nebenstehenden Beispiel generiert ein privilegiertes LLM (in Grün), ggf. nach vorhergehender Planung, Code, um E-Mail-Adressen auszulesen und E-Mails zu versenden.

Das nicht privilegierte Quarantäne-LLM (in Blau) wird von dem privilegierten LLM aufgerufen, um lediglich die relevanten textbasierten Datenfelder zu befüllen, ohne selbst Code generieren oder weitere Tools ausführen zu können. Die benötigten Daten stammen aus einer nicht vertrauenswürdigen Quelle (der E-Mail), welche möglicherweise Prompt-Injektionen enthält.

Constitutional Classifiers

Implementieren Sie spezialisierte Klassifikationsmodelle, die Ein- und Ausgaben in Echtzeit auf schädliche Inhalte prüfen. Ansätze wie die Constitutional Classifiers von Anthropic konnten die Erfolgsrate von Jailbreaks von 86 auf 4.4 Prozent reduzieren[15].

Warum herkömmliche Sicherheitstests nicht ausreichen

Wie in den vorherigen Abschnitten gezeigt, sind sowohl die Bedrohungslandschaft von KI-Applikationen als auch die erforderlichen Sicherheitsmassnahmen vielschichtig und komplex.

Genau deshalb reichen herkömmliche Schwachstellen-Scans und Applikationstests nicht aus, um die spezifischen Risiken dieser Systeme abzudecken. Sie sind auf etablierte Angriffsmuster wie SQL Injection, XSS, IDOR oder RCE ausgelegt und decken die komplexe, KI-spezifische Angriffslandschaft, welche KI-Applikationen mit sich bringen, nicht ab.

Die Risiken von KI-Applikationen sind grundlegend anders:

- **Prompt-Injektionen** basieren auf natürlicher Sprache und lassen sich nicht mit signaturbasierten Scans erkennen.
- **Agentische Schwachstellen** wie Goal Hijacking oder Memory Poisoning entstehen im Zusammenspiel von Modellverhalten, Orchestrierungslogik und Systemarchitektur.
- **Die Angriffsvektoren** sind hochdynamisch und erfordern kreative, manuelle Angriffssimulationen durch spezialisierte Penetration Tester mit Expertise in KI-Sicherheit.
- **Tool-Integrationen und MCP-Konfigurationen** müssen im Kontext der Gesamtarchitektur bewertet werden, nicht isoliert davon.

Ein spezialisierter KI Penetration Test schliesst diese Lücke. Er kombiniert automatisierte Angriffstechniken mit manueller Verifikation und deckt neuartige Schwachstellen auf, die herkömmliche Tests nicht erkennen können.

Automatisierte Angriffstechniken bilden dabei die erste Schicht: Spezialisierte Frameworks generieren systematisch Tausende von Angriffsvarianten und sondieren das Modell auf diverse Varianten von Prompt-Injektionen und Missbrauchspotential, wie zum Beispiel Jailbreaks.

Da automatisierte Tools jedoch primär bekannte Angriffsmuster abdecken, ist die manuelle Verifikation durch spezialisierte Penetration Tester unerlässlich. Sie bewerten die Ergebnisse im Kontext der konkreten Applikationsarchitektur, identifizieren logische Schwachstellen in der Orchestrierungsschicht und simulieren zielgerichtete Angriffe, die kein Scanner antizipiert.

Herkömmliche Schwachstellen-Scans erkennen weder Prompt-Injektionen noch unsichere Agenten-Logik oder manipulierte RAG-Pipelines. Diese Schwachstellen erfordern gezielte, manuelle Angriffssimulationen durch spezialisierte Penetration Tester.

Fazit und Empfehlungen

Die Bedrohungslandschaft für KI-Applikationen hat sich mit dem Aufkommen agentischer KI-Systeme grundlegend verändert. Die zentralen Erkenntnisse dieses Whitepapers lassen sich wie folgt zusammenfassen:

- **Agentische KI-Systeme schaffen eine völlig neue Kategorie von Sicherheitsrisiken, die über die vielzähligen bekannten LLM-Schwachstellen hinausgehen.** Tool Misuse, Agent Goal Hijacking, Memory Poisoning und Agent Communication Poisoning sind reale Bedrohungen mit potenziell gravierenden Auswirkungen. Daher sollten KI-Systeme als Mitarbeitende betrachtet werden, die besonders anfällig für Social-Engineering-Angriffe sind.
- **Prompt-Injektionen bleiben der gefährlichste Angriffsvektor und werden durch agentische Systeme massiv verstärkt.** Die meisten publizierten Verteidigungsmechanismen bieten kein ausreichendes Schutzniveau.
- **Das MCP-Protokoll als neuer Standard für Tool-Integrationen bringt spezifische Sicherheitsrisiken mit sich, die gezielt adressiert werden müssen.**
- **Herkömmliche Sicherheitstests reichen nicht aus, um die neuartigen Schwachstellen agentischer KI-Systeme aufzudecken. Dazu bedarf es spezialisierter KI Penetration Tests.**

Konkrete Handlungsempfehlungen

Bestandsaufnahme durchführen: Erfassen Sie alle KI-Applikationen in Ihrer Organisation sowie deren Tool-Zugriffe, Berechtigungen und Datenflüsse. Identifizieren Sie agentische Systeme mit hoher Autonomie als Priorität.

Least-Privilege-Prinzip umsetzen: Überprüfen Sie die Zugriffsrechte aller KI-Agenten und reduzieren Sie diese auf das erforderliche Minimum. Implementieren Sie granulare und dynamische Berechtigungskonzepte, die sich an den jeweiligen Auftraggeber und dessen Rechte anpassen. Statisch vergebene Privilegien bergen das Risiko, im falschen Kontext zu viel Zugriff zu gewähren. Kann ein Agent auf Daten zugreifen, auf welche die nutzenden Personen keinen Zugriff haben sollten, ist das ein Designfehler.

Human-in-the-Loop etablieren: Stellen Sie sicher, dass alle sicherheitskritischen Aktionen eine menschliche Bestätigung erfordern, bevor sie ausgeführt werden.

Alle Ein- und Ausgaben validieren, bereinigen und begrenzen: Überprüfen Sie sämtliche Ein- und Ausgaben auf Einhaltung definierter Schemata, bereinigen Sie Daten vor der Weitergabe an nachgelagerte Systeme und forcieren Sie realistische Token-Limits.

Spezialisierte Sicherheitstests beauftragen: Lassen Sie Ihre KI-Applikationen durch einen spezialisierten KI Penetration Test prüfen, der die neuartigen Angriffsvektoren abdeckt.

Schützen Sie Ihre KI-Applikationen

Minimieren Sie Risiken in Ihren KI-Anwendungen, bevor Angreifer sie ausnutzen. Oneconsult bietet spezialisierte KI Penetration Tests an, die Sicherheitslücken in Ihren KI-Anwendungen aufdecken, bevor Angreifer sie aktiv ausnutzen können.

AI Agent Security

Im Rahmen eines Grey-Box-Testansatzes analysieren wir Ihre KI-Applikationen einschliesslich Architekturunterlagen, Prompt-Templates, Tool-Spezifikationen und Konfigurationsoberflächen.

Ihre Vorteile:

- Sie erhalten eine ganzheitliche Sicherheitsbewertung, die sowohl die grafische Benutzeroberfläche als auch REST-API-Endpunkte und administrative Konfigurationspanels abdeckt.
- Sie erhalten einen umfassenden Bericht mit priorisierten Schwachstellen und konkreten Massnahmenvorschlägen.

› Jetzt mehr erfahren

Fragen zum Service oder interessiert an einem Angebot?

Kontaktieren Sie uns für eine individuelle Cybersecurity-Beratung.

Thomas Tunkel

Principal Client Services Manager

thomas.tunkel@oneconsult.com

linkedin.com/thomas-tunkel

Fachliche Fragen zu KI-Sicherheit?

Der Autor dieses Whitepapers steht Ihnen gerne zur Verfügung.

Dr. Marco Bertenghi

Senior Penetration Tester

marco.bertenghi@oneconsult.com

linkedin.com/bertenghi

Glossar

Agent Communication Poisoning	Manipulation der Kommunikation zwischen Agenten in Multi-Agent-Systemen, um Folgeanweisungen oder Ergebnisse zu verfälschen
Agentisches System	KI-System mit Planungsfähigkeit, Tool-Zugriff und autonomer Entscheidungsfindung
Context Leakage	Unbeabsichtigte Weitergabe sensibler Informationen aus dem Kontext eines Agenten an externe Systeme oder nachgelagerte Agenten
Context Rot	Schleichende Degradierung der Entscheidungsqualität eines Agenten durch Akkumulation veralteter, widersprüchlicher oder manipulierter Informationen im Kontext
Defense in Depth	Mehrschichtiger Sicherheitsansatz mit überlappenden Schutzmechanismen
DoW	Denial of Wallet – Kostenbasierter Angriff, bei dem durch massenhafte oder übermässig lange Anfragen gezielt hohe Inference-Kosten verursacht werden
Goal Hijacking	Manipulation eines Agenten, um ein anderes Ziel als vom Benutzer beabsichtigt zu verfolgen
IDOR	Insecure Direct Object Reference – Unbefugter Zugriff auf Ressourcen durch Manipulation von Referenzen
Jailbreak	Versuch, Sicherheitsmechanismen eines KI-Systems vollständig zu umgehen
KI (Englisch: AI)	Künstliche Intelligenz (Englisch: Artificial Intelligence) – Systeme, die neue (synthetische) Inhalte (Text, Code, Bilder, Audio, Video) erzeugen
Least Privilege	Sicherheitsprinzip, bei dem jede Komponente nur die minimal notwendigen Zugriffsrechte erhält
LLM	Large Language Model – Grosses Sprachmodell, das natürliche Sprache versteht und generiert
MCP	Model Context Protocol – Standardprotokoll für Tool-Integration in KI-Systemen
Memory Poisoning	Einschleusen schädlicher Daten in den persistenten Speicher eines Agenten
OWASP	Open Worldwide Application Security Project – Gemeinnützige Organisation für Anwendungssicherheit

Plan Manipulation	Angriff, bei dem die Planungslogik eines Agenten manipuliert wird, um die Ausführungsreihenfolge oder Zielpriorität zu verändern
Privilege Escalation	Erlangung höherer Zugriffsrechte als ursprünglich vorgesehen – bei Agenten oft durch die Ausnutzung transitiver Berechtigungen in der Aufrufkette; im Kontext von Agenten wird auch von Privilege Creep gesprochen
Prompt-Injektion	Angriff, bei dem Eingaben manipuliert werden, um das Verhalten eines KI-Systems zu verändern
RAG	Retrieval-Augmented Generation – Erweiterung eines LLM um externe Datenquellen
RCE	Remote Code Execution – Ausführung beliebigen Schadcodes auf einem Zielsystem aus der Ferne
Rogue Agent	Agentische KI-Komponente, die sich ausserhalb ihrer vorgesehenen Zielvorgaben verhält – durch Kompromittierung, Fehlkonfiguration oder emergentes Verhalten
Rug Pull	Angriff, bei dem ein zunächst legitim erscheinendes MCP-Tool oder ein externer Dienst nachträglich durch schädliche Funktionalität ersetzt wird
Sandboxing	Ausführung von Code oder Prozessen in einer isolierten Umgebung zur Schadensbegrenzung
Social Engineering	Manipulation von Personen (oder Agenten) durch psychologische Täuschung, um unberechtigten Zugang oder Informationen zu erlangen
SQL Injection	Einschleusen schädlicher SQL-Befehle in Datenbankabfragen zur Manipulation oder zum Auslesen von Daten
SSRF	Server-Side Request Forgery – Angriff, bei dem ein Server oder Agent dazu gebracht wird, Anfragen an interne oder unberechtigte Ressourcen zu senden
Tool Misuse	Missbrauch legitimer Werkzeuge eines Agenten für unbeabsichtigte oder schädliche Zwecke
Tool Poisoning	Angriff, bei dem manipulierte Tool-Beschreibungen versteckte Anweisungen an einen KI-Agenten enthalten
XSS	Cross-Site Scripting – Angriff zur Einschleusung schädlicher Skripte in Webanwendungen

Quellenverzeichnis

1. **OWASP Foundation.** (2025). OWASP Top 10 for LLM Applications 2025. https://genai.owasp.org/llm-top-10/

2. **Mairie, J.** (2026, 28. Januar). The Clawdbot Dumpster Fire: 72 Hours that Exposed Everything Wrong with AI Security. Acuvity. https://acuvity.ai/the-clawdbot-dumpster-fire-72-hours-that-exposed-everything-wrong-with-ai-security/

3. **OWASP Foundation.** (2026). OWASP Top 10 for Large Language Model Applications (v2.0). https://owasp.org/www-project-top-10-for-large-language-model-applications/

4. **OWASP Foundation.** (2026). OWASP Top 10 for Agentic Applications for 2026. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

5. **OWASP Foundation.** (2025). OWASP MCP Top 10. https://owasp.org/www-project-mcp-top-10/

6. **Nasr, M., et al.** (2025). The Attacker Moves Second: Stronger Adaptive Attacks Bypass Defenses Against LLM Jailbreaks and Prompt Injections. arXiv:2510.09023. https://arxiv.org/abs/2510.09023

7. **Herrador, M., Rehberger J.** (2026). SpAIware: Uncovering a novel artificial intelligence attack vector through persistent memory in LLM applications and agents. Future Generation Computer Systems. https://doi.org/10.1016/j.future.2025.107994

8. **Yang, X., et al.** (2026). Zombie Agents: Persistent Control of Self-Evolving LLM Agents via Self-Reinforcing Injections. arXiv:2602.15654. https://www.arxiv.org/abs/2602.15654

9. **Lupinacci, M., et al.** (2025). The Dark Side of LLMs: Agent-based Attacks for Complete Computer Takeover. arXiv:2507.06850. https://arxiv.org/abs/2507.06850

10. **Anthropic / MCP Community.** (2026). Model Context Protocol (MCP). Abgerufen am [Datum einfügen], von https://modelcontextprotocol.io

11. **Zou, W., et al.** (2025). PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. USENIX Security '25. https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag

12. **Anthropic.** (2025). Poisoning Attacks on LLMs Require a Near-constant Number of Poison Samples. arXiv:2510.07192. https://arxiv.org/abs/2510.07192

13. **Debenedetti, E., et al.** (2025). Defeating Prompt Injections by Design. arXiv:2503.18813. https://arxiv.org/pdf/2503.18813

14. **Willison, S.** (2023, 25. April). The Dual LLM Pattern for Building AI Assistants That Can Resist Prompt Injection. Simon Willison's Weblog. https://simonwillison.net/2023/Apr/25/dual-llm-pattern/

15. **Anthropic.** (2025). Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming. arXiv:2501.18837. https://arxiv.org/abs/2501.18837

Weiterführende Ressourcen

- OWASP LLM Top 10 for Large Language Model Applications: https://owasp.org/www-project-top-10-for-large-language-model-applications/
- OWASP Top 10 for Agentic Applications: https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
- OWASP MCP Top 10: https://owasp.org/www-project-mcp-top-10/
- MITRE ATLAS AI Security 101: https://atlas.mitre.org/resources/ai-security-101
- MITRA ATLAS Mitigations: https://atlas.mitre.org/mitigations