
# Referenzarchitektur

Eine typische Referenzarchitektur für Autonomous Red Teaming besteht aus sieben logisch getrennten Schichten, die zusammen ein kontrollierbares, skalierbares und kontinuierlich arbeitendes Gesamtsystem bilden.

**1. Steuerungs- und Governance-Schicht**

Auf der obersten Ebene befinden sich Policy Engine, Scope-Definition, Rollenmodell, Freigabemechanismen und Kill-Switch-Funktionen. Hier wird festgelegt, welche Umgebungen getestet werden dürfen, welche Aktionen zulässig sind und unter welchen Bedingungen ein Eingriff oder Abbruch erfolgen muss. Diese Schicht definiert den operativen Rahmen für alle nachgelagerten Komponenten.

**2. Orchestrierungs-Schicht**

Darunter liegt die zentrale Orchestrierung. Sie zerlegt übergeordnete Missionsziele in Teilaufgaben, priorisiert Angriffspfade, weist Aufgaben an spezialisierte Agenten zu und steuert deren Zusammenarbeit. Gleichzeitig verarbeitet sie Feedback aus der Zielumgebung und passt die laufende Missionsplanung dynamisch an.

**3. Agenten-Schicht**

In dieser Schicht arbeiten spezialisierte Agenten mit klaren Rollen. Typische Agententypen sind Reconnaissance-Agenten, Schwachstellenanalyse-Agenten, Exploitation-Agenten, LotL-Agenten und Agenten für laterale Bewegung. Jeder Agent verfügt über Wahrnehmungs-, Entscheidungs- und Aktionslogik für seinen Aufgabenbereich.

**4. Analyse- und Entscheidungs-Schicht**

Hier werden Telemetrie, Asset-Daten, Identitätsinformationen, Netzwerkbeziehungen und externe Threat-Intelligence zusammengeführt. KI-Modelle, regelbasierte Logik und Risikomodelle bewerten den Systemzustand, priorisieren Handlungsoptionen und leiten daraus adaptive Entscheidungen ab. Diese Schicht bildet das „kognitive Zentrum“ der Architektur.

**5. Digital-Twin-Schicht**

Ein zentrales Differenzierungsmerkmal ist der Digital Twin of the Infrastructure. Er modelliert Netzwerke, Identitäten, Trust-Zonen, kritische Systeme und Abhängigkeiten in einer simulierten Umgebung. Angriffspfade können dort vorbereitet, getestet und optimiert werden, bevor sie in der Live-Umgebung ausgeführt werden. Dadurch sinkt das operative Risiko und die Präzision steigt.

**6. Ausführungs-Schicht**

Diese Ebene enthält die technischen Aktionsmodule. Dazu gehören Scanner, API-Connectoren, Identitäts- und Berechtigungsanalysen, kontrollierte Exploit-Mechanismen sowie LotL-Funktionalitäten über PowerShell, Bash oder Cloud-APIs. Die Module setzen die Entscheidungen der Agenten technisch um, jedoch immer innerhalb der von Governance und Orchestrierung vorgegebenen Grenzen.

**7. Telemetrie-, Logging- und Audit-Schicht**

Parallel zu allen Schichten läuft die Erfassung von Logs, Ereignissen, Entscheidungen und Auswirkungen. Diese Daten dienen der Echtzeitüberwachung, der Risiko- und Impactbewertung, der Auditierbarkeit sowie der nachträglichen Analyse. Gleichzeitig speisen sie die Feedback-Loops, über die Agenten und Orchestrierung ihr Verhalten anpassen.

**Typischer Datenfluss**

Die Architektur arbeitet zyklisch: Telemetrie und Kontextdaten werden gesammelt, in der Analyse-Schicht ausgewertet, durch die Orchestrierung in Missionsentscheidungen überführt und an spezialisierte Agenten delegiert. Bevor riskante Aktionen im Live-System erfolgen, werden sie im Digital Twin validiert. Die Ergebnisse aus der Ausführung fließen wiederum in Logging, Monitoring und Feedback zurück und aktualisieren das Lagebild.

**Beispielhafte Minimalbesetzung in der Praxis**

Für einen ersten produktiven Einsatz genügt oft bereits ein reduziertes Modell: zentrale Policy Engine, ein Orchestrator, zwei bis drei Agententypen, ein Basis-Digital-Twin für Kernsysteme, standardisierte Telemetrie-Anbindung und ein revisionssicheres Logging. Mit wachsendem Reifegrad kann diese Architektur schrittweise um zusätzliche Agenten, feinere Entscheidungsmodelle und umfangreichere Simulationsfunktionen erweitert werden.

Dieses Beispiel zeigt, dass eine Referenzarchitektur nicht als starres Zielbild verstanden werden sollte, sondern als modularer Rahmen, der technische Leistungsfähigkeit, operative Kontrolle und regulatorische Anforderungen in ein belastbares Gesamtmodell überführt.