Holger Reibold

KI Red Teaming

Wie Organisationen KI-Risiken erkennen, testen und beherrschen

BRAIN-MEDIA.DE

Holger Reibold

KI Red Teaming

Wie Organisation KI-Risiken erkennen, testen und beherrschen

BRAIN-MEDIA.DE

ISBN: 978-3-95444-304-8

Cover: Freepik

Brain-Media.de
Dr. Holger Reibold – Hubert-Müller-Str. 52c – 66113 Saarbrücken

info@brain-media.de – www.brain-media.de

Inhaltsverzeichnis

Inhaltsverzeichnis I

Vorwort 1

1 Was ist KI-Red-Teaming? 5

1.1 Definition und Zielsetzung 5

1.2 Abgrenzung zu Application Security und ML Security 8

1.3 Typische Einsatzszenarien 9

2 Rechtlicher und ethischer Rahmen 11

2.1 Recht und Ethik als integrale Bestandteile 11

2.2 DSGVO-Grundlagen für KI-Systeme 12

2.3 Personenbezogene Daten in Prompts, Logs und Outputs 13

2.4 EU AI Act: Relevanz für KI-Red-Teaming 14

2.5 Haftung bei KI-Fehlverhalten 14

2.6 White-Hat-Ethik im KI-Kontext 15

3 Bedrohungsmodelle für KI-Systeme 17

3.1 Warum Threat Modeling notwendig ist 17

3.2 Klassische Bedrohungsmodelle: Was passt, was nicht? 18

3.3 STRIDE adaptiert für KI-Systeme 19

3.4 OWASP Top 10 for LLM Applications 20

3.5 Angriffsflächen neu denken 20
3.6 Trust Boundaries bei RAG- und Agentensystemen 22
3.7 Mini-Threat-Model: Beispiel Chatbot 23
4 Angriffsflächen jenseits des Prompts 27
4.1 Prompt Injection neu gedacht 27
4.2 Direkte vs. indirekte Prompt Injection 29
4.3 Klassische Jailbreak-Muster 31
4.4 Instruction Smuggling: Struktur transportiert Bedeutung 37
4.5 Wenn die KI „sieht“ und „hört“ 39
4.5.1 Visual Prompt Injection (VPI) 39
4.5.2 Audio-Jailbreaks und Voice-Cloning 40
4.5.3 Cross-Modale Datenlecks 40
4.5.4 Testing-Strategie für Multimodalität 40
4.6 Prompt Chaining und Rollenkonfusion 41
4.7 Bewertung der Risiken in realen Systemen 44
5 Mehrstufige Angriffe und kumulative Effekte 49
5.1 Warum Einzelschritt-Tests falsche Sicherheit erzeugen 49
5.2 Sequenzielle Eskalation ohne Regelbruch 51
5.3 Kumulative Effekte und semantische Drift 52
5.4 Verstärkung durch Automatisierung und Wiederholung 53
6 Model Manipulation und Poisoning 55

6.1 Überblick: Angriffe auf Modelle 55
6.2 Poisoning von Fine-Tuning-Daten 56
6.3 Manipulation von RAG-Datenquellen 57
6.4 Persistente Angriffe über Dokumente und Kontexte 58
6.5 Supply-Chain-Risiken bei Modellen 60
6.6 Grenzen klassischer Integritätsprüfungen 60
7 Agent-basierte Angriffe und Missbrauch 63
7.1 Was sind LLM-Agenten? 63
7.2 Tool Injection: Vom Prompt zur Aktion 65
7.3 Prompt → Code → Wirkung 66
7.4 Environment Poisoning 66
7.5 Privilegieneskalation durch Delegation 67
7.6 Worst-Case-Szenarien 68
8 Guardrails, Moderation und Policy Enforcement 69
8.1 Was Guardrails leisten – und was nicht 69
8.2 Input- vs. Output-Validierung 73
8.3 Regex und Keyword-Blocking 76
8.4 Prompt-Firewalls und Moderationsmodelle 79
8.5 Typische Failure-Modes von Guardrails 82
8.5.1 Failure-Mode 1: False Negatives 82
8.5.2 Failure-Mode 2: False Positives 83

8.5.3 Failure-Mode 3: Context Blindness 83
8.5.4 Failure-Mode 4: Automation Blindness 84
8.5.5 Failure-Mode 5: Governance Blindness 84
8.5.6 Failure-Mode 6: Kontrollillusion 85
8.6 Guardrails als Teil eines Sicherheitsmodells 86
9 Sichere Prompt- und Systemarchitektur 89
9.1 Least-Privilege-Prinzip für LLMs 89
9.2 Trennung von Prompts 93
9.3 Kontext-Minimierung statt -Maximierung 96
9.4 Output-Sanitization und Post-Processing 98
9.5 Referenzarchitektur für sichere LLM-Systeme 101
9.6 Human-in-the-Loop als Sicherheitskontrolle 103
10 Monitoring, Logging und Incident Response 105
10.1 Warum Logging bei LLMs kritisch ist 106
10.2 Was geloggt werden sollte – und was nicht 108
10.3 Anomalieerkennung in LLM-Nutzung 111
10.4 Prompt-basierte Angriffserkennung 112
10.5 Umgang mit Sicherheitsvorfällen 115
10.6 Lessons Learned 117
11 KI-Red-Teaming: strukturierter Prozess 119
11.1 Phase 1: Scope und Zieldefinition 120

11.2 Phase 2: Recon und Systemverständnis 122
11.3 Phase 3: Angriff und Exploration 125
11.4 Phase 4: Bewertung und Risikoanalyse 128
11.5 Phase 5: Reporting und Remediation 131
11.6 Checkliste für Red Teams 132
12 Fallstudie: Red-Teaming eines HR-Chatbots 135
12.1 Systembeschreibung und Annahmen 136
12.2 Threat Modeling des HR-Chatbots 139
12.3 Durchführung der Angriffe 144
12.4 Analyse der Ergebnisse 148
12.5 Risikoklassifizierung 151
12.6 Handlungsempfehlungen 154
12.7 Executive Summary 158
13 Lokales Testen ohne Cloud 161
13.1 Warum lokal testen? 161
13.2 Überblick lokaler LLM-Stacks 166
13.3 Setup eines lokalen Testlabors 169
13.4 Grenzen lokaler Tests 172
14 Hands-on – Annahmen unter Druck 175
14.1 Prompt Injection 176
14.2 Datenextraktion aus einem RAG-System 180

14.3 Agent-Hijacking über Tool-Use 184
14.4 Training Data Extraction 187
14.5 Komplettes KI-Sicherheitsaudit 191
Zum Schluss 195
Quellenverzeichnis VII
Stichwortverzeichnis IX
Mehr von Brain-Media.de XV

Vorwort

Im Frühjahr 2024 verkündete ein führendes Technologieunternehmen stolz, sein KI-System habe „menschliches Urteilsvermögen“ erreicht. Wenige Wochen später brachte ein 17-jähriger Schüler eben dieses System mit einer einzigen, vollständig regelkonformen Interaktion dazu, detaillierte Anleitungen für illegale Aktivitäten auszugeben.

Kein Bug.

Kein Exploit.

Kein Regelbruch.

Nur eine Annahme, die sich im realen Betrieb als falsch erwies.

Willkommen im Zeitalter des KI-Red-Teaming.

Während Sprachmodelle Gedichte schreiben, Geschäftsstrategien entwerfen und medizinische Zusammenfassungen liefern, bleibt eine unbequeme Wahrheit oft unbeachtet: KI-Systeme sind nicht intelligent im menschlichen Sinne – sie sind anschlussfähig. Sie reagieren auf Kontext, Erwartungen und Wahrscheinlichkeiten. Genau darin liegt ihre Stärke. Und genau darin liegt ihr Risiko.

KI-Systeme vertrauen jedem Input, weil sie nicht unterscheiden können, warum etwas gesagt wird. Sie reproduzieren Muster, weil sie keine eigene Bewertung von Angemessenheit besitzen. Sie halluzinieren mit hoher Überzeugungskraft, weil Plausibilität ihr Optimierungsziel ist –

nicht Wahrheit. Und sie wirken besonders überzeugend dort, wo menschliche Skepsis nachlässt.

Das Gefährliche daran ist nicht, dass KI-Systeme Fehler machen.

Das Gefährliche ist, dass sie korrekt funktionieren – unter Annahmen, die im realen Betrieb nicht mehr gelten.

Deshalb reicht es nicht aus, KI-Systeme einfach zu bauen und zu deployen. Es reicht nicht, sie mit Richtlinien zu versehen, Logs zu sammeln oder offensichtliche Regelverstöße zu verhindern. Wer KI produktiv einsetzt, muss sie prüfen, stressen und hinterfragen – nicht als Software, sondern als soziotechnisches System.

Dieses Buch ist kein dystopischer Alarmruf. Es ist ein Werkzeugkasten für Realisten.

Für Entwicklerinnen und Entwickler, die verstehen wollen, wie sich Modellverhalten im Nutzungskontext verändert.

Für Security- und Red-Teams, die erkennen, dass klassische Exploit-Logik bei KI-Systemen an ihre Grenzen stößt.

Für Compliance-, Risiko- und Datenschutzverantwortliche, die Sicherheit nicht mehr nur als Regelkonformität begreifen können.

Und für Entscheiderinnen und Entscheider, die wissen müssen, was „vertrauenswürdig“ bei KI tatsächlich bedeutet.

KI-Red-Teaming ist dabei mehr als das Testen von Prompt Injection oder Jailbreaks. Es ist eine Haltung. Eine Methode. Der Versuch, implizite Annahmen explizit zu machen – bevor sie zur Grundlage realer

Entscheidungen werden. Es geht nicht darum, KI-Systeme zu sabotieren, sondern darum, ihre Wirkungen unter realistischen Bedingungen sichtbar zu machen.

Die Angriffe, die in diesem Buch beschrieben werden, sind keine Heldengeschichten. Sie sind bewusst banal. Sie zeigen, wie regelkonforme Interaktionen, harmlose Kontexte und scheinbar vernünftige Outputs über Zeit, Skalierung und Automatisierung zu systemischem Schaden führen können. Nicht, weil jemand das System „gehackt" hat, sondern weil es zu sehr vertraut wurde.

Am Ende geht es nicht darum, KI zu stoppen.

Und auch nicht darum, sie perfekt zu machen.

Es geht darum, sie entscheidbar zu machen.

Und damit verantwortbar.

Viel Freude – und die nötige Skepsis – auf dieser Reise.

Holger Reibold

(Januar 2026)

1 Was ist KI-Red-Teaming?

KI-Red-Teaming ist ein Begriff, der in den letzten Jahren inflationär verwendet wurde. Er bezeichnete Prompt-Spielereien, automatisierte Testläufe, Bug-Bounty-Programme für Sprachmodelle oder schlicht alles, was sich kritisch mit KI-Systemen beschäftigte. Diese Unschärfe ist kein Zufall. Sie ist Ausdruck eines Übergangs: Klassische Sicherheitskonzepte reichen nicht mehr aus, neue sind noch nicht etabliert. Dieses Kapitel schafft Klarheit – nicht durch Abgrenzung aus Prinzip, sondern durch eine präzise funktionale Definition.

1.1 Definition und Zielsetzung

KI-Red-Teaming ist keine Sammlung von Angriffstechniken, sondern eine sicherheitsanalytische Methode. Um die weitere Diskussion auf eine solide Basis zu stellen, wird hier folgende Definition entwickelt:

KI-Red-Teaming ist die systematische Überprüfung der Annahmen, unter denen ein KI-System betrieben wird, indem plausible Schadenspfade unter realistischen Nutzungsbedingungen simuliert und bewertet werden.

Diese Definition ist bewusst nüchtern. Sie vermeidet Begriffe wie „Hack", „Exploit" oder „Breach", weil sie für KI-Systeme nur begrenzte Aussagekraft besitzen. Der Fokus liegt nicht auf dem Wie eines Angriffs, sondern auf dem Warum eines möglichen Schadens. Die Zielsetzung von KI-Red-Teaming ist nicht, ein System zu kompromittieren, sondern

Unsicherheit zu reduzieren. Es liefert Entscheidungsgrundlagen für Organisationen, keine Garantien für Sicherheit.

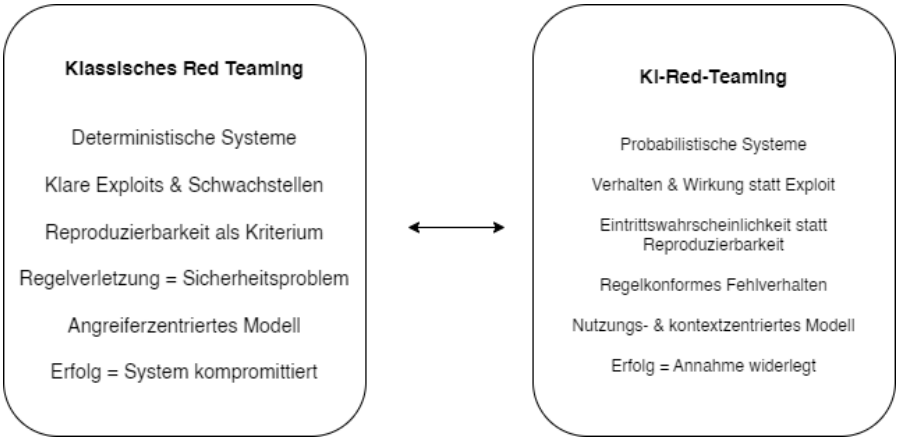
Ein erfolgreiches KI-Red-Teaming beantwortet Fragen wie folgende:

- Unter welchen Annahmen funktioniert dieses System sicher?
- Was passiert, wenn diese Annahmen im Betrieb nicht gelten?
- Welche Wirkungen entstehen dann realistisch?

Klassisches Penetration Testing basiert auf einem klaren Sicherheitsmodell: Ein System gilt als sicher, solange es nicht erfolgreich kompromittiert werden kann. Der Fokus liegt auf technischen Schwachstellen, klaren Angriffsvektoren und reproduzierbaren Exploits. Dieses Modell funktioniert gut für deterministische Systeme.

Für KI-Systeme greift es zu kurz.

Ein KI-System kann vollständig regelkonform betrieben werden, keine Schwachstelle im klassischen Sinne enthalten – und dennoch Schaden verursachen. Es gibt keinen „Exploit“, keinen klaren Eintrittspunkt, keinen Moment der Kompromittierung. Stattdessen entsteht Risiko durch korrektes Verhalten unter falschen Annahmen.



Vergleich der zugrunde liegenden Annahmen klassischer Red-Teaming-Ansätze mit den Eigenschaften produktiver KI-Systeme. Während traditionelles Red Teaming auf deterministische Systeme, reproduzierbare Exploits und klare Regelverletzungen ausgerichtet ist, adressiert KI-Red-Teaming probabilistisches Verhalten, kontextabhängige Wirkung und die Prüfung impliziter Annahmen.

KI-Red-Teaming unterscheidet sich deshalb grundlegend:

- Es sucht keine Schwachstellen, sondern Annahmen.
- Es bewertet Wirkung, nicht nur Verhalten.
- Es akzeptiert Nicht-Determinismus als gegeben.

Penetration Testing fragt: „Wie breche ich das System?“

KI-Red-Teaming fragt: „Wann wird das System gefährlich?“

Beides ist notwendig – aber nicht austauschbar.

1.2 Abgrenzung zu Application Security und ML Security

Application Security fokussiert die sichere Entwicklung und den Betrieb von Software. ML Security wiederum adressiert spezifische Risiken maschineller Lernverfahren: Trainingsdaten, Modellintegrität, Adversarial Examples. KI-Red-Teaming liegt quer zu beiden Disziplinen.

Es betrachtet weder ausschließlich die Anwendung noch ausschließlich das Modell, sondern das Zusammenspiel von Modell, Daten, Nutzung, Prozessen und Organisation. Viele sicherheitsrelevante Effekte entstehen genau dort, wo Zuständigkeiten verschwimmen.

Ein formal robustes Modell kann in einer schlecht eingebetteten Anwendung hochriskant sein. Eine sichere Anwendung kann durch falsche Nutzung eines Modells Schaden verursachen. KI-Red-Teaming schließt diese Lücke. Es ersetzt weder AppSec noch ML Security, sondern ergänzt sie um eine systemische Perspektive.

In klassischen Sicherheitsmodellen sind Rollen klar verteilt: Red Teams greifen an, Blue Teams verteidigen. KI-Red-Teaming verschiebt dieses Verständnis. Das Red Team agiert hier nicht gegen Entwickler oder Systeme, sondern gegen implizite Gewissheiten. Es stellt Annahmen infrage, die im Alltag selten explizit formuliert werden:

- Nutzer verstehen die Grenzen des Systems
- Outputs werden kritisch geprüft
- Automatisierung verändert die Wirkung nicht
- Verantwortung bleibt klar zugeordnet

Das Blue Team sichert weiterhin technische Komponenten, überwacht Betrieb und reagiert auf Vorfälle. Das Purple Team übernimmt im KI-Kontext eine besonders wichtige Rolle: Es übersetzt Red-Team-Erkenntnisse in organisatorische Entscheidungen. KI-Red-Teaming ist kein Wettbewerb zwischen Teams; es ist ein kooperativer Erkenntnisprozess.

1.3 Typische Einsatzszenarien

KI-Red-Teaming entfaltet seine größte Wirkung dort, wo KI-Systeme produktiv eingesetzt werden und reale Entscheidungen beeinflussen. Typische Szenarien sind folgende:

- Chatbots mit Kunden- oder Mitarbeiterkontakt
- RAG-Systeme mit Zugriff auf interne Dokumente
- Entscheidungsunterstützung in sensiblen Domänen
- Agenten mit Tool- oder Systemzugriff
- Automatisierte Workflows mit geringer menschlicher Kontrolle

Je höher der Automatisierungsgrad und je größer die Reichweite eines Systems, desto relevanter wird KI-Red-Teaming. Nicht weil diese Systeme „unsicherer" sind, sondern weil ihre Fehlannahmen schneller und weiter wirken. Allerdings ist KI-Red-Teaming kein Schutzmechanismus. Es verhindert keine Angriffe, es patcht keine Systeme und es garantiert keine Sicherheit. Seine Stärke liegt in der Explizierung von Risiken, nicht in ihrer Eliminierung. Es kann aufzeigen, wo Annahmen falsch sind,

welche Schadenspfade plausibel sind und welche Entscheidungen daraus folgen sollten.

Was KI-Red-Teaming nicht leisten kann:

- Vollständige Abdeckung aller Nutzungsszenarien
- Vorhersage aller zukünftigen Missbrauchsformen
- Technische Absicherung gegen alle Risiken

Diese Grenzen sind kein Mangel, sondern eine Voraussetzung für Ehrlichkeit. KI-Red-Teaming verspricht nicht Kontrolle, sondern Erkenntnis.

Fassen wir zusammen: KI-Red-Teaming ist keine neue Spielart klassischer Sicherheitstests, sondern eine Antwort auf ein neues Sicherheitsproblem: Systeme, die korrekt funktionieren und dennoch schaden können. Die folgenden Kapitel verschieben den Fokus schrittweise von der Definition zur Methode, von der Methode zu den Angriffsflächen und von den Angriffsflächen zu realistischen Schadenspfaden.

Ab hier geht es nicht mehr um Begriffe. Ab hier geht es um Annahmen – und was passiert, wenn sie nicht gelten.

Quellenverzeichnis

Anthropic (2023). Anthropic red team reports. Online: https://www.anthropic.com/news/red-team-reports.

Bundesamt für Sicherheit in der Informationstechnik (2023). Leitfaden: Sichere Entwicklung von KI-Systemen. Online: https://www.bsi.bund.de/DE/Themen/Kuenstliche-Intelligenz/ki_node.html.

Carlini, N., et al. (2021). Extracting training data from large language models. USENIX Security Symposium.

Center for Research on Foundation Models (2023). CRFM red team reports. Stanford University. Online: https://crfm.stanford.edu/red-team.html.

Derczynski, L. (2023). Garak: A vulnerability scanner for large language models. Online: https://github.com/leondz/garak.

European Union (2024). The EU Artificial Intelligence Act - Up-to-date developments and analyses of the EU AI Act. Online: https://artificialintelligenceact.eu/.

Fraunhofer IAIS (2023). KI-Testfelder. Online: https://www.iais.fraunhofer.de/de/kompetenzen/kuenstliche-intelligenz/ki-testfelder.html.

Google DeepMind (2023). Red teaming AI systems. Online: https://deepmind.google/discover/blog/red-teaming-ai-systems/.

Majumdar, S., et al. (2025). Red Teaming Large Language Models. arXiv. https://arxiv.org/abs/2509.11398.

Microsoft (2021). Adversarial machine learning threat matrix. MITRE ATLAS. Online: https://atlas.mitre.org/.

National Institute of Standards and Technology (2023). AI Risk Management Framework (AI RMF 1.0). Online: https://www.nist.gov/itl/ai-risk-management-framework.

OWASP Foundation (2023). OWASP Top 10 for large language model applications. Online: https://owasp.org/www-project-top-10-for-large-language-model-applications/.

Perez, E., Ringer, S., Gao, K., et al. (2022). Red teaming language models with language models. arXiv. https://arxiv.org/abs/2202.03286.

Shokri, R., et al. (2017). Membership inference attacks against machine learning models. IEEE Symposium on Security and Privacy.

UK AI Safety Institute (2024). AI Safety Institute red teaming standard. UK Government. Online: https://www.gov.uk/government/publications/ai-safety-institute-red-teaming-standard.

Stichwortverzeichnis

A

Adversarial Examples 8
Agent 63
Agent-basierte Angriffe 63
Agent-Hijacking 184
Aktionsebene 150
Angriff 125
Angriffsdurchführung 144
Angriffserkennung 112
Angriffsfläche 20, 27
Annahmen 7, 136
Anomalieerkennung 111
API 60
Application Security 8
AppSec 8
Artefakte 15
Audio-Jailbreak 40
Automation Blindness 84
Automatisierung 53
Autoritätseskalation 142

B

Bedrohungsmodell 17
Bewertung 119, 128
Blacklist 73
Blue Team 8
Bug 1

C

Chatbot 9
Checkliste 132
Cloud 161
Compliance 172
Context Blindness 83
Context Injection 41
Context-Angriff 34
Contextual Leakage 141
Cross-Modal 40
Cross-Modal Jailbreaking 41
CVSS 47, 128

D

Datenextraktion 180
Datenminimierung 12
Datenschutzgrundverordnung 12
Definition 5
Denial of Service 19
Direkte Prompt Injection 30
Dokumentation 125

DSGVO 12

E

Einsatzszenarien 9
Elevation of Privilege 19
Empfehlungsebene 150
Entscheidungsebene 150
Entscheidungsunterstützung 9, 172
Environment Poisoning 66
Ergebnis 148
Essenz 25
Ethik 11
EU AI Act 14
Explizierung 9
Exploit 1
Exploitability 128
Exploration 125

F

Failure-Mode 82
False Negative 77, 82
False Positive 77, 83
Fehlverhalten 141
Filter 33
Fine-Tuning-Daten 56
Foundation Model 60
Fragmentierung 32

G

Governance Blindness 84
Guardrails 40, 69

H

Haftung 14
Handlungsempfehlungen 154
HR-Chatbot 135
Human-in-the-Loop 103

I

Impact 128
Incident Response 105, 115
Indirekte Prompt Injection 30
Information Disclosure 19
Informationsebene 149
Informationspreisgabe 141
Inhaltsanalyse 113
Input-Validierung 73
Instruction Smuggling 37
Integrität 12
Integritätsprüfung 60

J

Jailbreak 31

K

Katalysator 51
Keyword-Blocking 76
Keyword-Filter 73
KI-Output 45
KI-Red-Teaming 1
KI-Sicherheitsaudit 191
Klassifikation 130
Klassifikationsmodell 73
Kompromittierung 6
Kontext 20
Kontext-Minimierung 95, 96
Kontextualisierung 32
Kontextverwaltung 137
Kontrollillusion 85
Kumulative Effekte 52

L

Lab 175
Latenzzeit 50
Least Privilege 95
Least Significant Bits 41
Least-Privilege-Prinzip 89
LLM 17
LLM-Agent 63
LLM-Stack 166
Logging 105
Lokales Testen 161

M

Manipulation 57
Mehrstufige Angriffe 49
Metadaten 28, 107
Mini-Threat-Model 23
ML Security 8
Model Manipulation 55
Moderationsmodell 80
Monitoring 105
Multimodalität 40
Multi-Turn 31
Muster 1
Mustererkennung 73

N

Nachvollziehbarkeit 119
Nutzerverhalten 36
Nutzungskontext 2

O

Output-Nutzung 137
Output-Sanitization 98
Output-Validierung 73
OWASP Top 10 20

P

Penetration Testing 6

Persistente Angriffe 58
Personenbezogene Daten..... 13
Plausibilität 1
Poisoning..... 55
Policy 74
Post-Processing..... 98
Privilegieneskalation..... 67
Prompt 66
Prompt Chaining 41, 44
Prompt Injection.....27, 29, 176
Prompt-Architektur 89
Prompt-Firewall..... 79
Purple Team 9

R

RAG 9, 22
Rechenschaftspflicht..... 12
Recht 11
Recon 122
Red Team..... 8
Red-Teaming-Ansatz..... 7
Referenzarchitektur 101
Regelpriorisierung 33
Regex 76
Rekonstruktion 145
Remediation 119, 131
Reporting 131
Reproduzierbarkeit..... 119
Repudiation 19
Richtlinie 2
Risikoanalyse 128
Risikobewertung..... 151
Risikoklassifizierung 151
Risikomatrix 24
Rollenkonfusion..... 44
Rollenumkehr..... 32

S

Schadenspfad 21
Scope..... 15, 120
Scope-Definition 15
Scoring-Modell 47
Semantische Drift 52
Sequenzielle Eskalation 51
Sicherheitsfilter..... 40
Sicherheitsmodell..... 6, 86
Sicherheitsvorfall..... 115
Skalierbarkeit 148
Spoofing..... 19
Steganografie-Check 41
STRIDE 19
Strukturierter Prozess..... 119
Supply-Chain-Risiken 60
Systemarchitektur 89
Systembeschreibung 136
Systemprompt 97
Systemverständnis..... 122

T

Tampering 19
Testlabor 172
Threat Modeling 17, 139
Tool Injection 65
Training Data Extraction 187
Trust Boundaries 22

U

Überzeugungskraft 1

V

Verhalten 7
Vertraulichkeit 12
Visual Prompt Injection 39
Voice-Cloning 40

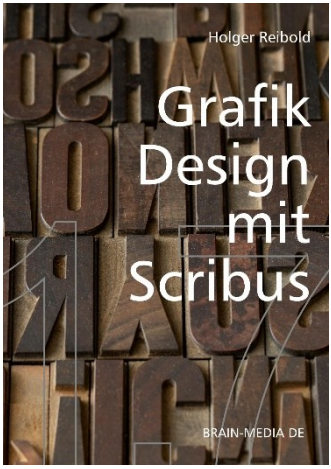
W

White-Hat-KI-Red-Teaming 15
Whitelist 73
Wiederholung 53
Wirkpfad 46
Wirkung 148
Wirkungsebene 129
Wirkungsintegrität 61
Workflow 9
Worst-Case-Szenarien 68

Z

Zieldefinition 120
Zielsetzung 5
Zweckbindung 12

Mehr von Brain-Media.de



Grafikdesign mit Scribus

In diesem Handbuch erfahren Sie alles, um mit Scribus ein professionelles Projekt umzusetzen – angefangen bei der Entwicklung kreativer Ideen bis zur konkreten Gestaltung.

Preis: 24,99 EUR

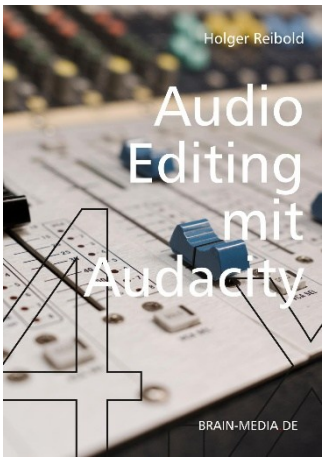
Umfang: 420 Seiten

Virtuelle Maschinen mit VirtualBox 7.x

So verwandeln Sie einen Rechner in ein ganzes Netzwerk oder bauen ein Testumgebung auf. Dieses Handbuch führt Sie in alle wichtigen Funktionen bis hin zur Cloud-Nutzung ein.

Preis: 16,99 EUR

Umfang: 150 Seiten



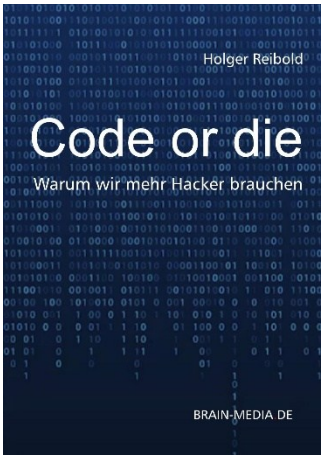
Audio Editing mit Audacity 4.x

Alles Wichtige, was Sie für den erfolgreichen Einsatz des freien Audioeditors wissen müssen.

Umfang: 220 Seiten

Preis: 19,99 EUR

Erscheint: Frühjahr 2026



Code or die – Warum wir mehr Hacker brauchen

Ein Manifest für mehr digitale Selbstbestimmung, Neugierde und Eigenverantwortung. Medienkompetenzen alleine genügen nicht; die Gesellschaft von morgen braucht Digitalkompetenzen.

Umfang: 120 Seiten

Preis: 14,99 EUR

Erscheint Frühjahr 2026

Private KI – KI-Systeme lokal betreiben, kontrollieren und verantworten

Alles Wichtige für den sicheren Einsatz von lokalen KI-Systemen.

Umfang: 140 Seiten

Preis: 16,99 EUR

Erscheint: Frühjahr 2026

KI Incident Response – Wie man Sicherheitsvorfälle in KI-Systemen erkennt, eindämmt und verantwortet

Ziel- und punktgenaue Reaktionen für kritischen KI-Vorfälle.

Umfang: 140 Seiten

Preis: 16,99 EUR

Erscheint: Frühjahr 2026