(Don‘t) Believe the Hype: Schwachstellenanalyse mit KI

14. Oktober 2026

Who am I?

Dipl.-Inf. Matthias Deeg
Senior Expert IT Security Consultant
Head of Research & Development
CISSP, CISA, OSCP, OSCE

- Seit frühen Tagen an Informationstechnologie interessiert – insbesondere an IT-Sicherheit
- Informatikstudium an der Universität Ulm, Deutschland
- IT Security Consultant seit 2007

Agenda

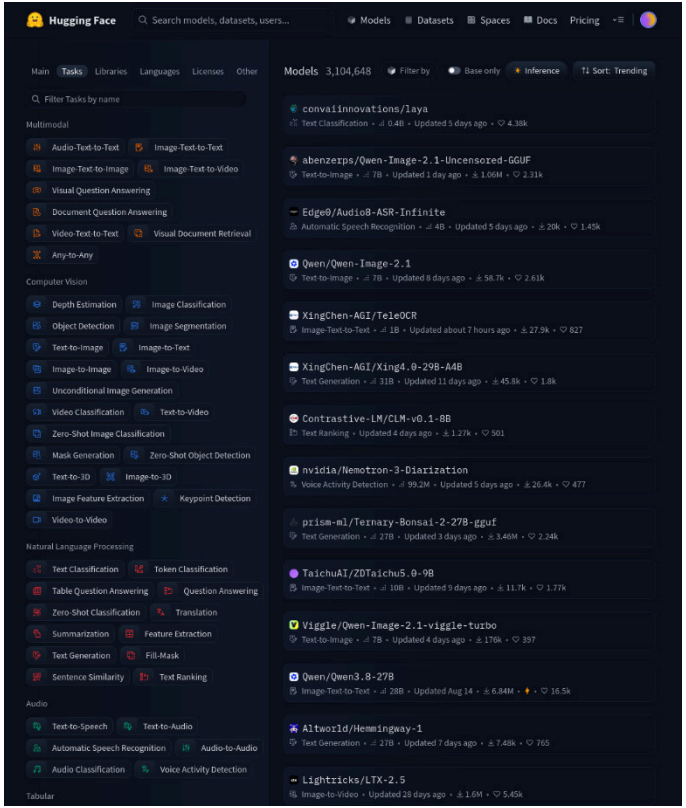
1. Künstliche Intelligenz
2. KI-Agenten
3. Schwachstellenanalysen
4. Fallstudie: DICOM Softwaretools
5. Trends
6. Fazit
7. Fragen & Antworten

Künstliche Intelligenz

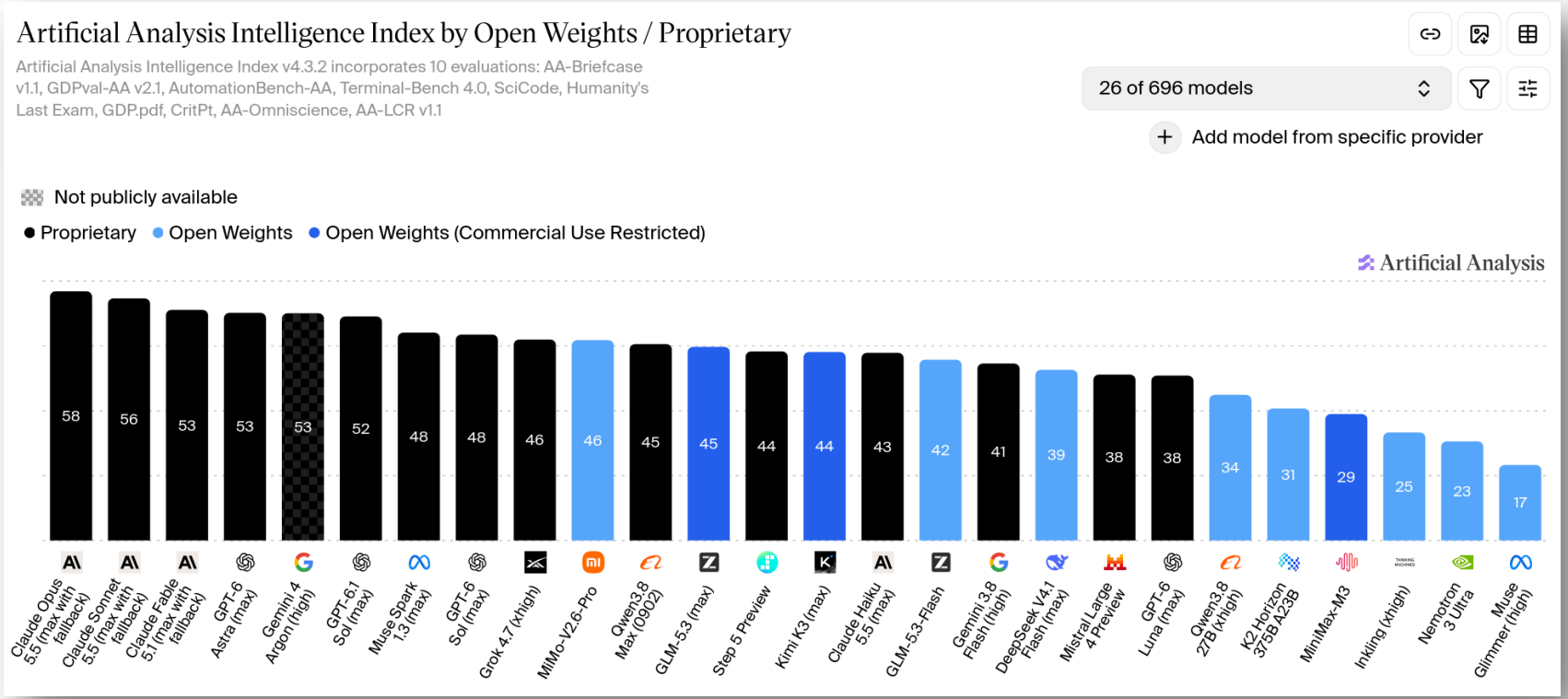
- Im Rahmen dieses Vortrags beschränkt sich der Begriff künstliche Intelligenz auf Transformer-basierte Large Language Models (LLMs)
- Eigenschaften:
  - Tokenisierung
  - Vorhersage des nächsten Tokens (Next Token Prediction)
  - Pre-Training
  - Post-Training
  - Inferenz mit Kontext und Tools

Künstliche Intelligenz

- KI-Modelle für fast jeden Zweck, z. B.
  - Natural Language Processing
  - Computervision
  - Audio Processing
- Verschiedene Modalitäten:
  - Text
  - Bild
  - Audio
  - Video
- Open Source, Open Weight, Closed Source (AI-as-a-Service)



Künstliche Intelligenz



Quelle: https://artificialanalysis.ai/

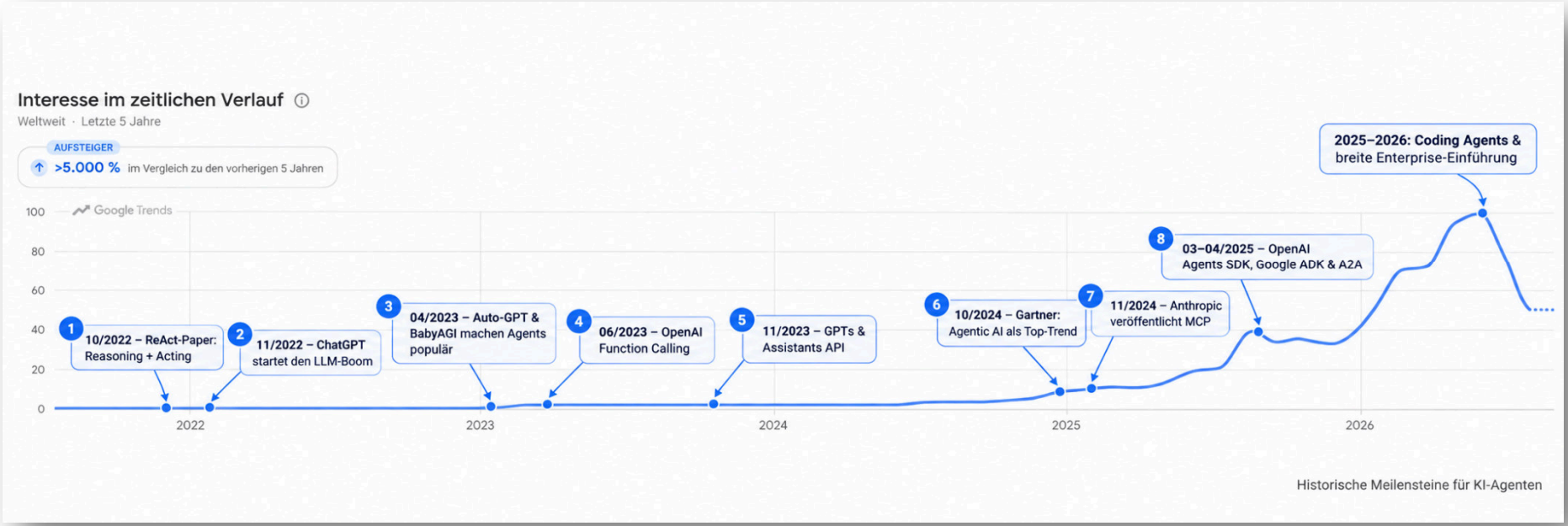
KI-Agenten

- Ein KI-Agent ist ein Softwaresystem, das ein Ziel erhält, seine nächsten Schritte selbst auswählt und hierfür bei Bedarf Werkzeuge nutzt.
- Typisch sind die folgenden vier Komponenten:
  1. Ziel: Der Agent versucht ein definiertes Ziel zu erfüllen.
  2. Entscheidung: Der Agent entscheidet selbst, was als Nächstes zu tun ist.
  3. Tool-Nutzung: Der Agent verwendet bei Bedarf verfügbare Werkzeuge zur Zielerfüllung, z. B. Datenbanken, Websuche, Terminal, Programmierumgebungen.
  4. Rückkopplung: Der Agent prüft Ergebnisse seiner Aktionen und passt basierend darauf sein weiteres Vorgehen an.

KI-Agenten: Relevante Ereignisse

Zeitpunkt	Ereignis
Oktober 2022	ReAct-Paper
November 2022	ChatGPT
April 2023	AutoGPT / BabyAGI
Juni 2023	OpenAI „Function Calling“
November 2023	Agenten mit „Tool Calling“ als Plattformfunktion bei OpenAI, Anthropic, Google
Oktober 2024	Anthropic „Computer Use“
November 2024	Model Context Protocol (MCP)
Dezember 2024	Devin (AI Coding Agent)
März 2025	OpenAI Agents SDK
April 2025	Google Agent Development Kit, Agent2Agent
September 2025	Claude Agent SDK
Januar 2026	OpenClaw (zuvor Clawd/Clawdbot/Moltbot)

KI-Agenten: Suchtrend



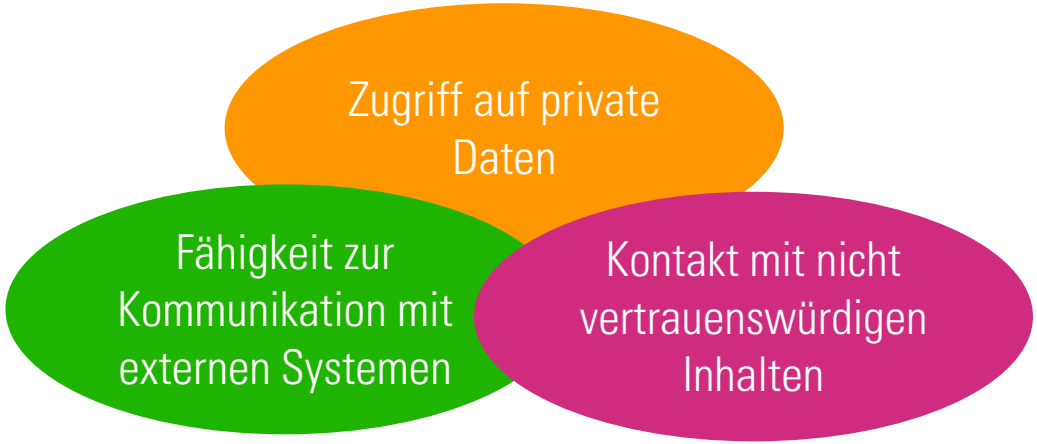
Google Trends Websuche „ai agent“ (weltweit)

KI-Agenten: Harness

- KI-Agent = KI-Modell + Rüstzeug/Gerüst (Harness / Scaffolding)
- Wichtige Komponenten eines Agent Harness:
  - Aufgabenverwaltung (Task Management)
  - Kontextverwaltung (Context Management)
  - Gedächtnis (Memory)
  - Werkzeugaufruf (Tool Calling)
  - Protokollierung (Logging/Tracing)
- Häufig Supervisor/Orchestrator Design Pattern
  - Delegierung von Aufgaben an Subagenten
  - Parallele Bearbeitung verschiedener Teilaufgaben

KI-Agenten: Problem

- *The Lethal Trifecta for AI Agents*
- KI-Agenten können Anweisungen und Regeln missachten oder anders interpretieren



https://simonw.substack.com/p/the-lethal-trifecta-for-ai-agents

KI-Agenten: Lethal Trifecta

(Illustratration von ChatGPT)

KI-Agenten & Frameworks

- Zahlreiche KI-Agenten und Frameworks für unterschiedliche Zwecke

Beispiele für KI-Agenten
Claude Code
Codex
Copilot
Gemini CLI
Muse Code
OpenCode
OpenClaw
Pi
Hermes Agent

Beispiele für KI-Frameworks
Open AI Agents SDK
Claude Agent SDK
Google Agent Development Kit
Microsoft Agent Framework
LangChain / LangGraph
NVIDIA-labs Object Oriented Agents (NOOA)
CrewAI
Pocket Flow
smolagents

Schwachstellenanalyse mit KI

https://openai.com/index/introducing-aardvark/

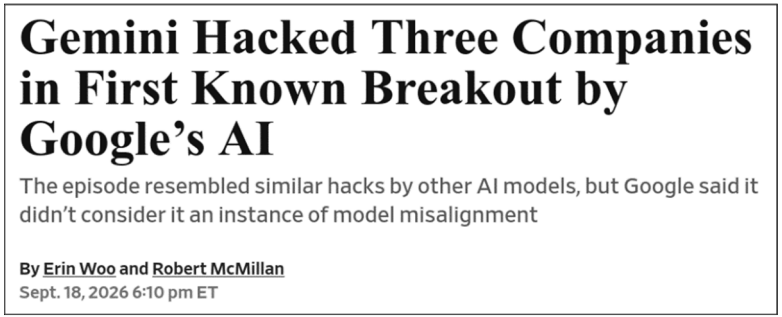
https://deepmind.google/blog/introducing-codemender-an-ai-agent-for-code-security/

https://xbow.com/blog/top-1-how-xbow-did-it

https://www.anthropic.com/research/zero-days

Schwachstellenanalyse mit KI

https://openai.com/index/hugging-face-model-evaluation-security-incident/



https://www.wsj.com/tech/ai/gemini-hacked-three-companies-in-first-known-breakout-by-googles-ai-5c0baba2

https://www.anthropic.com/news/investigating-incidents-cybersecurity-evals

Schwachstellenanalyse mit KI

- Intention: Automatisiertes Finden von Sicherheitsschwachstellen
- Klassische Schutzziele der Informationssicherheit
  1. Vertraulichkeit (Confidentiality)
  2. Integrität (Integrity)
  3. Verfügbarkeit (Availability)
- Weitere Schutzziele
  - Authentizität (Authenticity)
  - Zurechenbarkeit (Accountability)
  - Nichtabstreitbarkeit (Non-repudiation)

Schwachstellenanalyse mit KI

- MITRE Common Weakness Enumeration (CWE)
  - Software Development
  - Hardware Design
  - Research Concepts
- Common Vulnerabilities & Exposures (CVE)
- OWASP Top 10
- Verkettung von Schwachstellen (Exploit Chains)

1435 - Weaknesses in the 2025 CWE Top 25 Most Dangerous Software Weaknesses

- Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') - *(79)*
- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') - *(89)*
- Cross-Site Request Forgery (CSRF) - *(352)*
- Missing Authorization - *(862)*
- Out-of-bounds Write - *(787)*
- Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') - *(22)*
- Use After Free - *(416)*
- Out-of-bounds Read - *(125)*
- Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') - *(78)*
- Improper Control of Generation of Code ('Code Injection') - *(94)*
- Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') - *(120)*
- Unrestricted Upload of File with Dangerous Type - *(434)*
- NULL Pointer Dereference - *(476)*
- Stack-based Buffer Overflow - *(121)*
- Deserialization of Untrusted Data - *(502)*
- Heap-based Buffer Overflow - *(122)*
- Incorrect Authorization - *(863)*
- Improper Input Validation - *(20)*
- Improper Access Control - *(284)*
- Exposure of Sensitive Information to an Unauthorized Actor - *(200)*
- Missing Authentication for Critical Function - *(306)*
- Server-Side Request Forgery (SSRF) - *(918)*
- Improper Neutralization of Special Elements used in a Command ('Command Injection') - *(77)*
- Authorization Bypass Through User-Controlled Key - *(639)*
- Allocation of Resources Without Limits or Throttling - *(770)*

https://cwe.mitre.org/data/definitions/1435.html

Schwachstellenanalyse mit KI

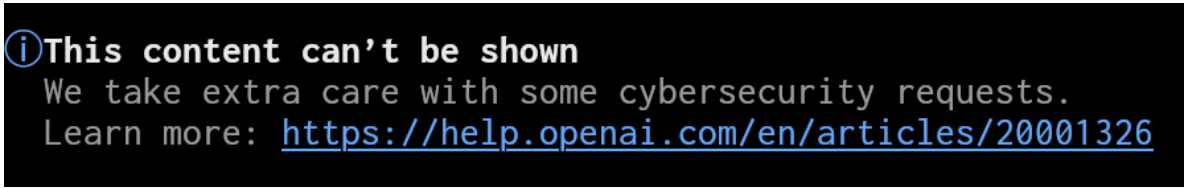
- Verschiedene Anwendungsgebiete, z. B.
  - IT/OT-Infrastruktur
  - Web-Applikationen
  - Web Services
  - Softwareanwendungen
  - Embedded Devices (Software, Firmware und Hardware)
- Verschiedene Szenarien und Perspektiven, z. B.
  - Informationsbasis: Black Box, Grey Box, White Box
  - Initialer Zugang: Innentäter, Außentäter
- Kombination von deterministischen und nicht-deterministischen Prozessen

Schwachstellenanalyse mit KI: Alignment

- AI Alignment: Ausrichtung des Verhaltens eines KI-Modells an gewünschten Zielen, Regeln und Werten
- Ziele, Regeln und Werte von unterschiedlichen Personen oder Gruppen können jedoch sehr verschieden sein
- Guardrails schränken KI-Modelle entsprechend ein, z. B.
  - Eingaben und Ausgaben werden auf unerlaubte Inhalte geprüft und ggf. angepasst
  - Unerwünschte Tool-Aufrufe werden blockiert
  - Aktionen werden nur nach berechtigter Freigabe zugelassen (Human-in-the-Loop)
- Oftmals spezielle Freigaben von Hersteller- bzw. Anbieterseite notwendig, wenn bestimmtes Verhalten genutzt werden soll, z. B. Cybersecurity Skills

Schwachstellenanalyse mit KI: Alignment

- Wer entscheidet über das Alignment eines KI-Modells?
  - der Hersteller
  - der Anbieter
  - der Nutzer
  - die Gesellschaft / der Gesetzgeber



- Open Weight-Modelle bieten die Möglichkeit, deren Verhalten anzupassen
- Mittels Fine-Tuning und Abliteration kann neues Verhalten antrainiert und Verweigerungen von Benutzeranfragen abtrainiert werden

Schwachstellenanalyse mit KI: Methode

- Allgemeine Vorgehensweise
  1. Übersicht über Testgegenstand verschaffen
  2. Bedrohungsmodell erstellen
  3. Hypothesen aufstellen
  4. Aufgaben erstellen
  5. Aufgaben bearbeiten (Hypothesen verifizieren/falsifizieren)
  6. Ergebnisse verifizieren
  7. Dokumentation erstellen
- Sicherheitsanalysen laufen nicht strikt sequenziell ab; iterative Teilprozesse (z. B. neue Hypothesen, Aktualisierung des Bedrohungsmodells)

Schwachstellenanalyse mit KI: Agenten

- Zahlreiche KI-Agenten, Harnesses, MCP Server und Skills für Sicherheitstests
- Kommerziell und frei, open source, z. B.
  - XBOW
  - Aikido
  - Strix
  - Shannon
  - PentAGI
  - PentestGPT
  - Cloudflare Security Audit Skills
  - Trail of Bits Skills

Fallstudie

- Testobjekte: DICOM Open-source Softwareprojekte
  1. DICOM ToolKit (DCMTK)
  2. Grassroots DICOM (GDCM)
  3. dcm4che
- Schwachstellenanalyse mit verschiedenen KI-Agenten
  - Claude Code
  - Hermes Agent
  - Eigenentwicklung (Ponos)
- Nutzung von kommerziellen closed-source und open-weight KI-Modellen

Fallstudie

- Fokus auf DICOM-Parser (Digital Imaging and Communications in Medicine)
- CLI-Tools und Programmbibliotheken
- Verwendung unterschiedlicher Programmiersprachen
  - C/C++: DICOM ToolKit (DCMTK) und Grassroots DICOM (GDCM)
  - Java: dcm4che

Methode

- Lokale KI-Agenten für Schwachstellenanalyse
  1. Claude Code
  2. Hermes Agent
  3. Ponos (R&D AI Agent)
- Lokales KI-Modell: Qwopus 3.6-27B-v2 (veröffentlicht Mitte Mai 2026)
- Zugriff auf Quellcode der Zielsoftware
- Zugriff auf lauffähige Zielsoftware
- Zugriff auf lokal installierte Softwaretools und Skills
  - Keine MCP-Tools
  - *„Bash is all you need“*
- „/goal“

Demo

- AI Coding Agent Claude Code mit lokalem KI-Modell (Qwen3.8-27B-Uncensored-FP8), Tools, Skills und einem Ziel

```
/goal Analyze this codebase for more security
vulnerabilities. Verify all findings by
creating PoC exploits stored in the exploits
directory. Document all findings in ANALYSIS.md
```

Ergebnis

- 16 automatisiert gefundene und verifizierte Sicherheitsschwachstellen mit hohem oder mittlerem Sicherheitsrisiko
- Diese 16 Schwachstellen manuell im Anschluss überprüft
- Security Advisories erstellt und im Rahmen unseres Responsible Disclosure-Prozesses an die betroffenen Hersteller bzw. Maintainer gemeldet
- KI-Agenten haben noch mehr Sicherheitsschwachstellen gefunden und dokumentiert, die bisher nicht weiter untersucht wurden
- Große Varianz bei Laufzeiten und Anzahl gefundener Schwachstellen pro Durchlauf

Ergebnis

- Ausgesuchte Testgegenstände wurden bereits zuvor auf Sicherheitsschwachstellen hin untersucht, auch mit KI
- Im Rahmen des R&D-Projekts konnten dennoch weitere Schwachstellen mit hohem und mittlerem Risiko gefunden werden („continue“)

Security Researcher Identifies Quintet of Bugs in Toolkit Used in DICOM Medical Imaging Software

Posted By Steve Alder on Jul 1, 2026

Gemeldete Sicherheitsschwachstellen (1/3)

- Autonom gefundene Schwachstellen in DICOM ToolKit (DCMTK) v3.7.0
- Innerhalb weniger Stunden behoben

#	Produkt	Schwachstellentyp	SySS ID	CVE ID	Duplikat
1	DICOM ToolKit (DCMTK)	Integer Overflow or Wraparound (CWE-190)	SYSS-2026-046	-	nein
2	DICOM ToolKit (DCMTK)	Path traversal (CWE-22)	SYSS-2025-047	-	nein
3	DICOM ToolKit (DCMTK)	Integer Overflow or Wraparound (CWE-190)	SYSS-2025-048	-	nein
4	DICOM ToolKit (DCMTK)	Integer Overflow or Wraparound (CWE-190)	SYSS-2025-049	-	nein
5	DICOM ToolKit (DCMTK)	Integer Overflow or Wraparound (CWE-190)	SYSS-2025-050	-	nein
6	DICOM ToolKit (DCMTK)	Command Injection (CWE-78)	SYSS-2025-051	-	ja
7	DICOM ToolKit (DCMTK)	Path traversal (CWE-22)	SYSS-2025-052	-	ja

Gemeldete Sicherheitsschwachstellen (2/3)

- Autonom gefundene Schwachstellen in Grassroots DICOM (GDCM) v3.3.0
- Bisher keine Rückmeldung zur Behebung der Schwachstellen erhalten

#	Produkt	Schwachstellentyp	SySS ID	CVE ID	Duplikat
1	Grassroots DICOM (GDCM)	Stack-based Buffer Overflow (CWE-121)	SYSS-2026-067	-	-
2	Grassroots DICOM (GDCM)	Stack-based Buffer Overflow (CWE-121)	SYSS-2025-068	-	-
3	Grassroots DICOM (GDCM)	Integer Overflow or Wraparound (CWE-190)	SYSS-2025-069	-	-
4	Grassroots DICOM (GDCM)	Integer Overflow or Wraparound (CWE-190)	SYSS-2025-070	-	-
5	Grassroots DICOM (GDCM)	Format String (CWE-134)	SYSS-2025-071	-	-

Gemeldete Sicherheitsschwachstellen (2/3)

- Autonom gefundene Schwachstellen in dcm4che v5.35.0
- Bisher keine Rückmeldung zur Behebung der Schwachstellen erhalten

#	Produkt	Schwachstellentyp	SySS ID	CVE ID	Duplikat
1	dcm4che	Path Traversal (CWE-22)	SYSS-2026-077	-	-
2	dcm4che	Path Traversal (CWE-22)	SYSS-2025-078	-	-
3	dcm4che	Path Traversal (CWE-22)	SYSS-2025-079	-	-
4	dcm4che	Path Traversal (CWE-22)	SYSS-2025-080	-	-

Beispiel: Ponos

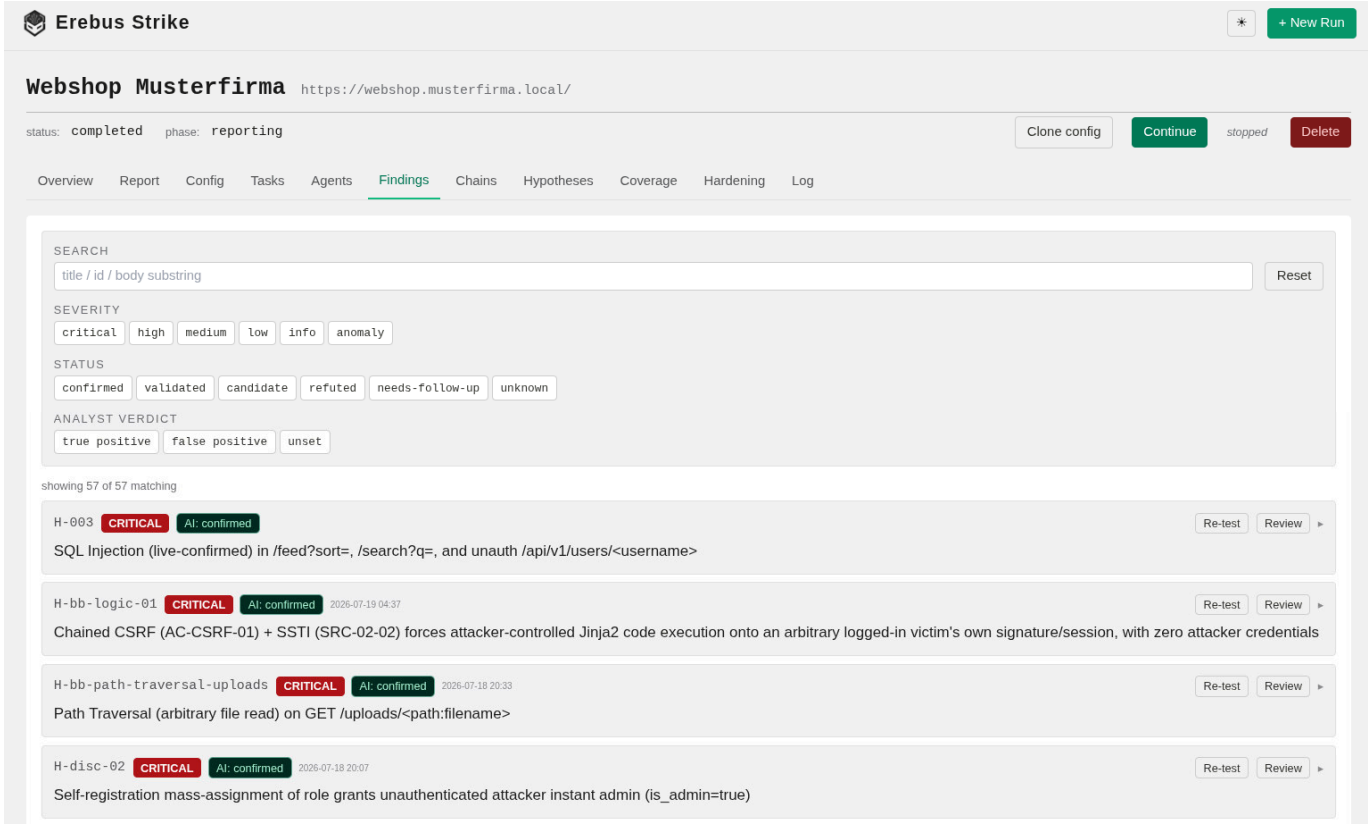
```
DCMTK 3.7.0 security review complete in /home/matt/cysecmed/result

Pipeline: threat model -> 10 hypotheses (H1-H10) -> dynamic/static testing + PoCs -> adversarial validation -> final
report (report/final_report.md). 12 finding files; 6 validated (findings/validated/):
- H1 HIGH CWE-78: storescp --exec-on-reception/-on-store command injection (unauthenticated RCE as service user) —
PoC proved
- F4 MEDIUM CWE-36/59: predictable store filename overwritten via peer-controlled symlink — PoC proved
- F6 MEDIUM CWE-532: cleartext password leaked in -v logging (dcuserid.cc:362-369) — PoC proved
- F7 MEDIUM CWE-22: getscu --bit-preserving writes peer SOPInstanceUID unsanitized (../ escape) — PoC proved
- F8 LOW-MEDIUM CWE-777: async exec hooks fork without wait (fan-out unless --exec-sync) — PoC proved
- H8 LOW CWE-190: DCM PixelData API accepts 64->32-bit truncated frame length (0xffffffff) then crashes — API-level
PoC (validator-confirmed)

Refuted: H2 (storescp path traversal), H4 (C-FIND SQL injection — this build has no SQL backend: flat DCMQRDBM index,
no sqlite linkage, no SQL construction in dcmqrdb).
Unconfirmed (kept in findings/): H3 (C-PRINT viewer, static-only, severity lowered to LOW), H5/H6 (JPEG/JLS decoder
memory safety — no crash on corpus; ASan/valgrind blocked by ~1GB per-process vmem cap), H9 (fork-per-association DoS
— mechanism confirmed static, bounded by
MaxAssociations=16; dynamic burst blocked by vmem cap), H10 (dcmtls — TLS off by default, no hardcoded weak floor;
deployment-dependent, LOW).

Adversarial validation: all 12 statuses consistent with evidence; adjustments applied (H3 MEDIUM->LOW, H8 ->
CONFIRMED API-level). PoC logs: logs/poc_F-4/F-6/F-7/F-8*.log all SUCCESS. Top recommendations: execve-argv/quote
exec hooks, atomic store-file creation, mask -v password,
sanitize getscu output path, keep frame length 64-bit end-to-end, re-test H3/H5H6 on an ASan-capable host.
⏱ Agent ran 9956.94 seconds
```

Beispiel: Erebus Strike



Trends: Flare-On Challenge 13

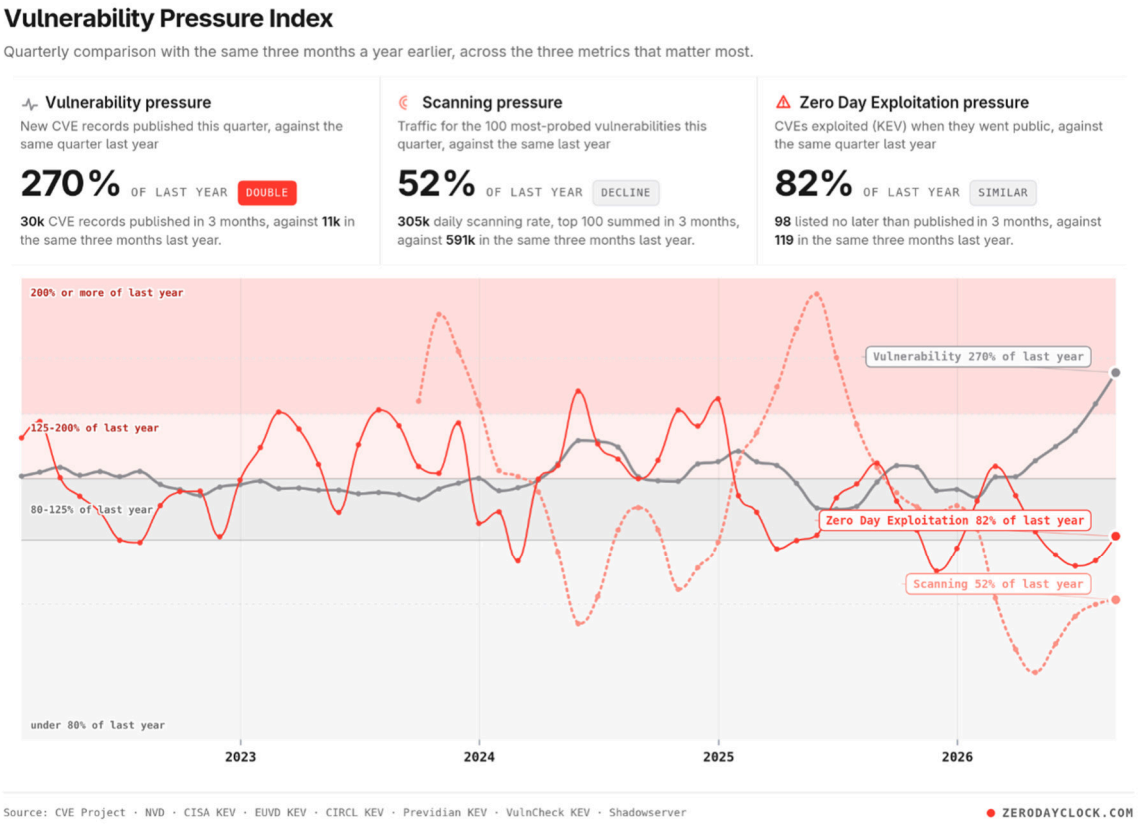
- Reverse Code Engineering CTF mit 9 Aufgaben
- Schnellster Teilnehmer: 1 Stunde, 5 Minuten und 43 Sekunden
- Fallstudie mit Codex und GPT-6.1-Sol
- Linux VM
- Lokal installierte Tools
- *„Bash is all you need“*

Trends: Flare-On Challenge 13

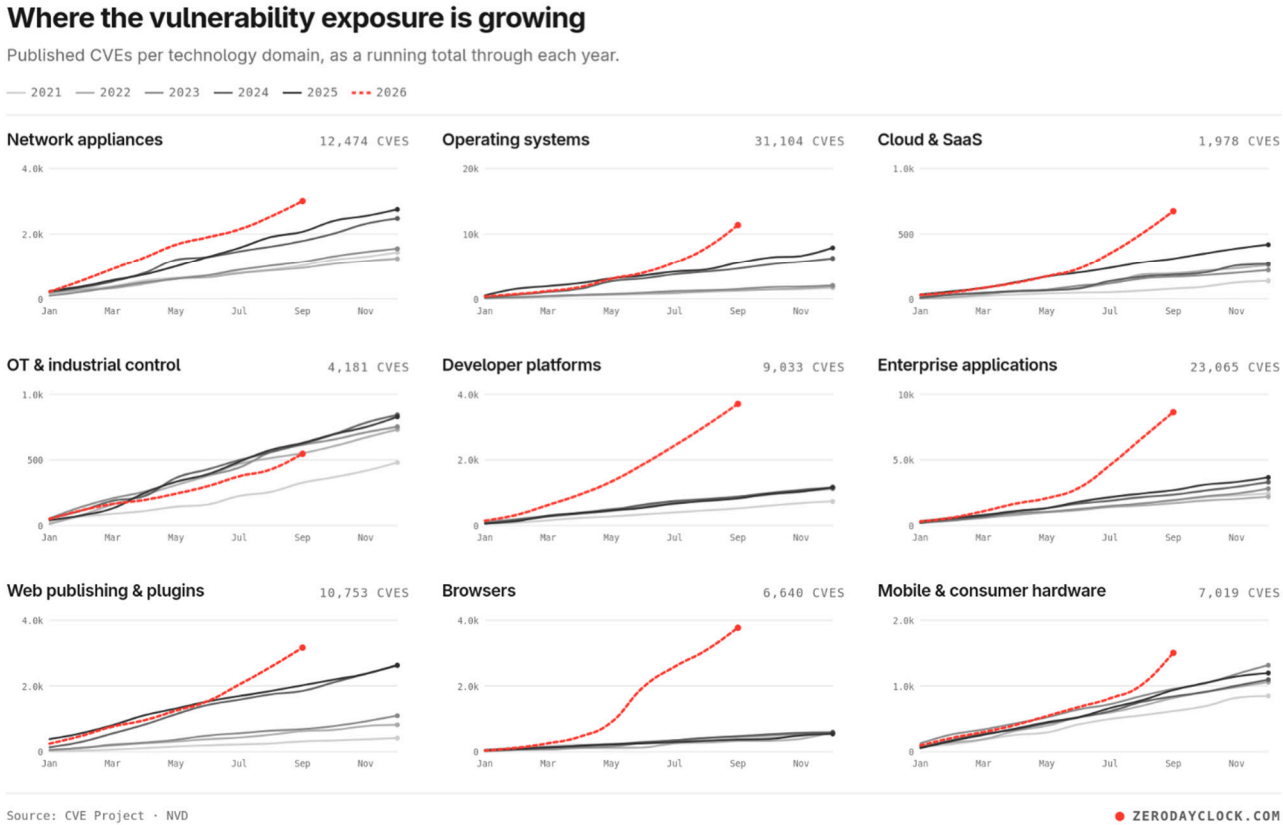
Challenge #	Zeit (Codex mit GPT-6.1-Sol)	Zeit (Hermes Agent mit abl. Qwen3.8-27B)
1	00:01:16	00:01:17
2	00:09:15	-
3	00:11:02	00:50:30
4	00:09:23	00:29:00
5	00:05:30	00:46:49
6	00:09:45	00:40:18
7	00:24:33	-
8	00:13:16	-
9	00:20:04	-

Trends: Hackbots

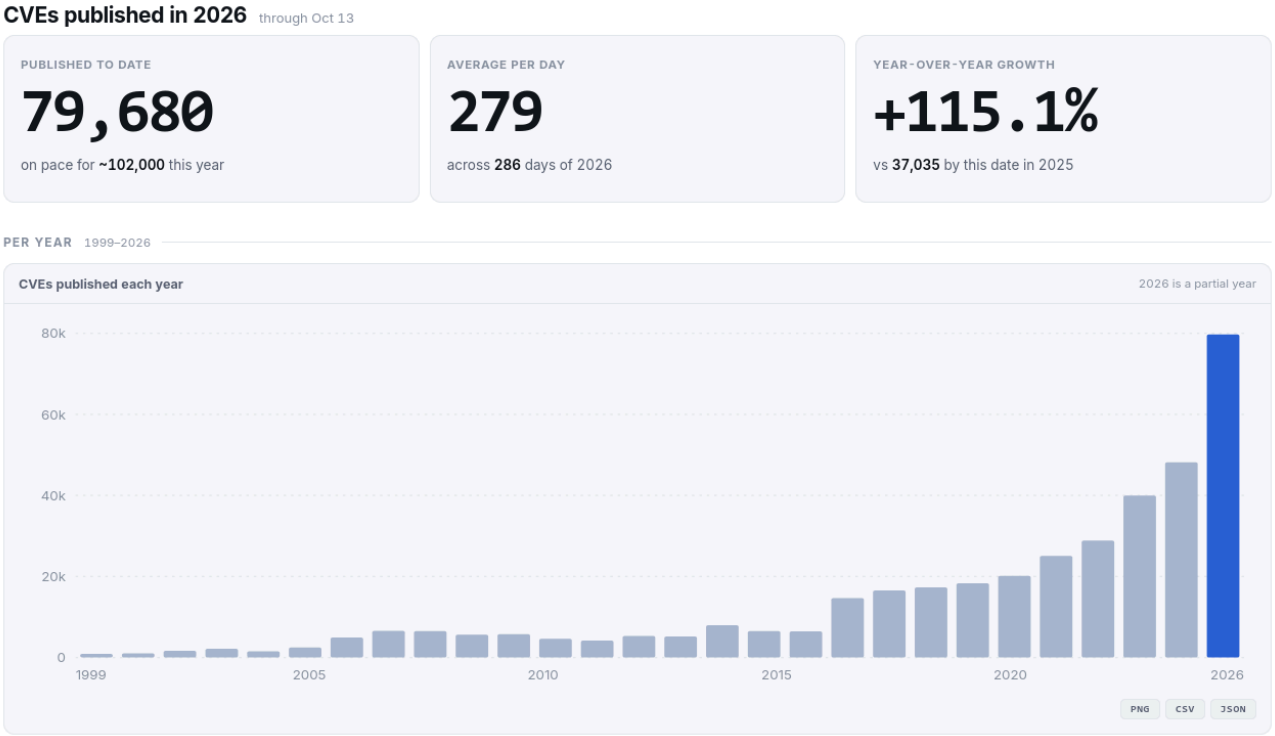
Trends: Schwachstellen



Trends: Schwachstellen



Trends: Schwachstellen



Quelle: https://cve.icu/

Trends: Schwachstellen

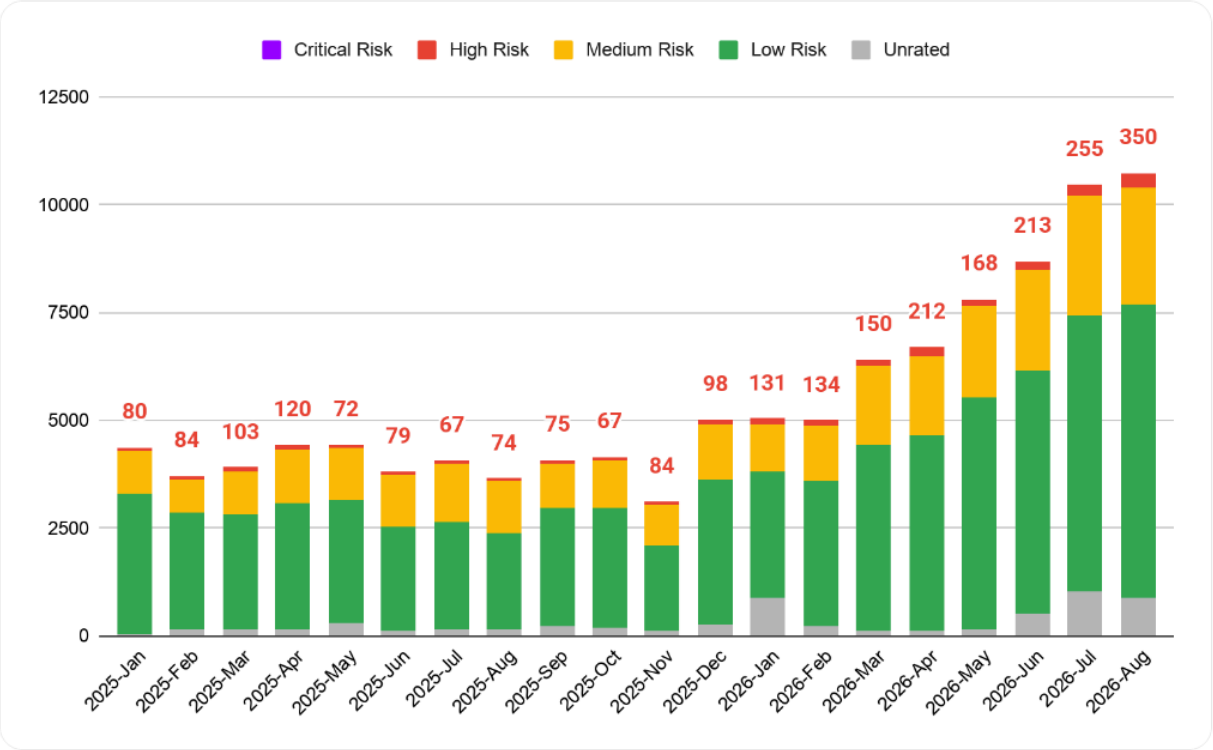
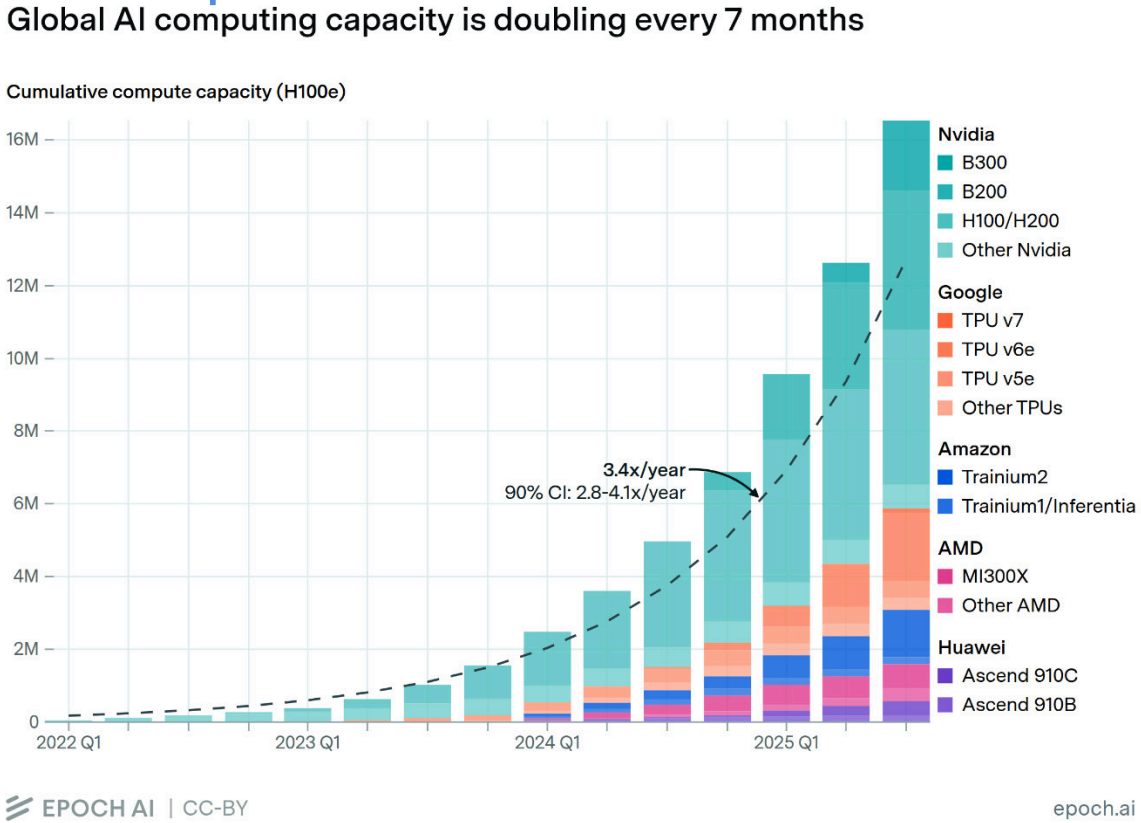
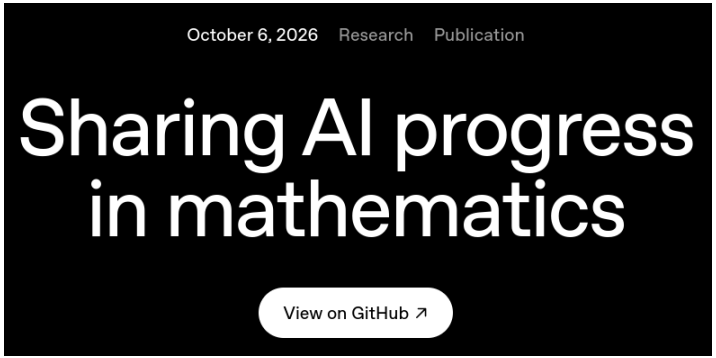


Figure 2: Count of vulnerabilities disclosed by GTIG vulnerability risk rating, January 2025 - August 2026 (Source: GTIG)

Trends: AI Compute

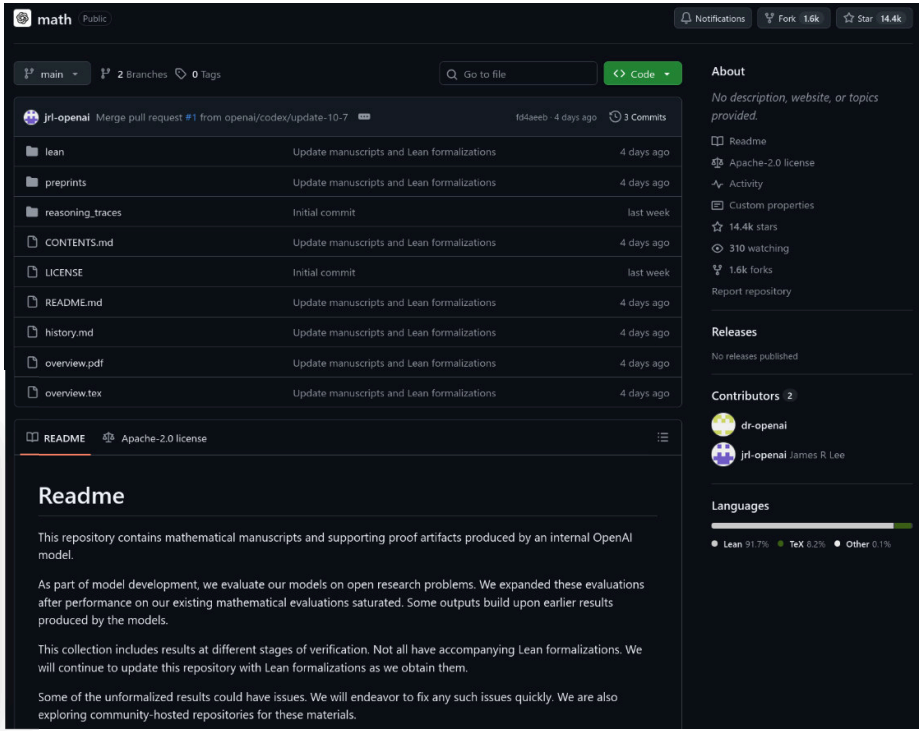


Trends: Veröffentlichung von Ergebnissen



https://openai.com/index/sharing-ai-progress-in-mathematics/

https://www.wired.com/story/openai-is-pissing-off-a-bunch-of-mathematicians-again/



https://github.com/openai/math

Trends: Mehr Tools

Fazit (1/2)

- Aktuelle agentische KI-Systeme sind in der Lage automatisiert Schwachstellen in IT-Systemen, z. B. Softwareanwendungen zu finden, zu verifizieren und auch auszunutzen
- Automatisierung beschleunigt viele Security-Aufgaben
- Fähigkeiten von KI-Modellen nehmen stetig zu
- Zuvor nützliche Funktionen von Harnessen können bei neuen KI-Modellen nutzlos oder gar kontraproduktiv werden

Fazit (2/2)

- Menschliches Verständnis für KI-erzeugte Ergebnisse zu schaffen kann aufwändig oder gar unmöglich sein (z. B. fehlende überprüfbare Belege)
  - Was wurde herausgefunden?
  - Wie wurde es herausgefunden?
  - Warum existiert ein Ergebnis?
  - Welche Bedeutung hat Ergebnis?
- Der Hype um künstliche Intelligenz hat seine Daseinsberechtigung
- Wir leben in spannenden Zeiten mit spannenden Herausforderungen und ungelösten Problemen

Vielen Dank …

… für Ihre Aufmerksamkeit!

Haben Sie Fragen?
Wie fanden Sie den Vortrag?

E-Mail: matthias.deeg@syss.de
Web: https://www.syss.de
Blog: https://blog.syss.com
YouTube: https://www.youtube.com/c/SySSPentestTV

Referenzen

1. Artificial Analysis Intelligence Index, https://artificialanalysis.ai/
2. ReAct: Synergizing Reasoning and Acting in Language Models, https://arxiv.org/abs/2210.03629, 2022
3. AutoGPT, https://github.com/Significant-Gravitas/AutoGPT
4. BabyAGI, https://github.com/yoheinakajima/babyagi
5. Introducing Aardvark: OpenAI's agentic security researcher, https://openai.com/index/introducing-aardvark/, 2025
6. Evaluating and mitigating the growing risk of LLM-discovered 0-days, https://www.anthropic.com/research/zero-days, 2026
7. DICOM ToolKit (DCMTK), https://dcmtk.org
8. Grassroots DICOM (GDCM), https://sourceforge.net/projects/gdcm/
9. dcm4che, https://web.dcm4che.org/
10. SYSS-2026-046, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-046.txt, 2026
11. SYSS-2026-047, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-047.txt, 2026
12. SYSS-2026-048, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-048.txt, 2026
13. SYSS-2026-049, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-049.txt, 2026
14. SYSS-2026-050, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-050.txt, 2026
15. SYSS-2026-067, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-067.txt, 2026
16. SYSS-2026-068, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-068.txt, 2026
17. SYSS-2026-069, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-069.txt, 2026
18. SYSS-2026-070, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-070.txt, 2026

Referenzen

19. SYSS-2026-071, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-071.txt, 2026
20. SYSS-2026-077, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-077.txt, 2026
21. SYSS-2026-078, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-078.txt, 2026
22. SYSS-2026-079, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-079.txt, 2026
23. SYSS-2026-080, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2026-080.txt, 2026
24. Vulnerability Pressure Index, https://zerodayclock.com/
25. Vulnerability Discovery and Exploitation Trends in the AI Era, Google Threat Intelligence Group, https://cloud.google.com/blog/topics/threat-intelligence/vulnerability-discovery-and-exploitation-trends-in-the-ai-era, 2026
26. Global AI computing capacity is doubling every 7 months, https://epoch.ai/data-insights/ai-chip-production, 2026

THE PENTEST EXPERTS

WWW.SYSS.DE