

Pentesting in the age of AI

Where are we?





*I'm not an AI-Guru-Prompt-Graph-
Harnes-Loop-Hype-Engineer*

About: me

- **Christian Biehler** – IT-Security Expert
 - Master in IT Security
 - Working since 2012 as hacker, pentester, consultant
 - More than 400 projects regarding the topics
 - **Pentesting** / Vulnerability Management
 - Network security
 - Building and maintaining ISMS
 - Windows-, Linux-, Web Security
 - Trainer for Web Security, Windows Security, Microsoft 365 Security, AI assisted security testing
 - Author and speaker for different (German) papers and conferences
 - Trained and certified
 - **ISO 27001** Lead Implementer, **ISO 27035** Incident Manager, **ISO 27005** Risk Manager
 - Certified Information Systems Security Professional (CISSP)
 - Certified Professional Penetration Tester (eCPPT)

Agenda

Today we talk about: AI

Introduction

AI doing security
assessments

Fundamentals

Overview

SAST

DAST



Fundamentals

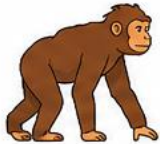
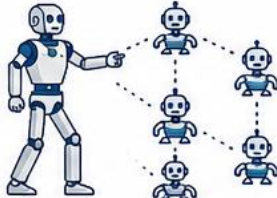
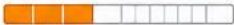
*AI is like your **four-years-old** child!*

Treat it accordingly:

- *When asking if your child is able to put on his shoes – don't wonder if nothing happens*
- *When asking AI if you can eat the mushroom ... you all know that GPT meme, right?*

THE EVOLUTION OF AI USE: FROM CHAT TO AUTONOMOUS MULTI-AGENT SYSTEMS

An analogy to human evolution – with opportunities, risks and degree of automation

EVOLUTION STAGE	1. Basic AI Access AI as Information Source	2. First Steps AI as Assistant	3. Leveraging AI AI as Co-pilot	4. Collaboration AI as Team Member	5. Delegation AI as Manager	6. Autonomy AI as Multi-Agent System
						
	Simple questions, quick answers	Supports individual tasks & content creation	Actively helps with creation, analysis & synthesis	Takes over partial tasks in processes, works together with you	Plans, coordinates and monitors tasks – you set goals	Multiple AI agents work autonomously in networks, make decisions & act
ANALOGY	You ask – AI answers. Like discovering fire.	You ask – AI helps. Like first steps of humankind.	You work – AI supports. Like developing skills and expanding abilities.	You work together – on eye level. Like teamwork with other people.	You set goals – AI implements. Like a project manager you can trust.	AI agents operate autonomously in a network. Like a society that organizes itself.
AUTOMATION LEVEL	0–10% 	10–30% 	30–60% 	60–80% 	80–95% 	95–100% 
OPPORTUNITIES	✓ Faster access to information ✓ Ideas & inspiration ✓ Easy to use	✓ Time savings ✓ Better texts & results ✓ Productivity boost	✓ Higher quality ✓ Creativity & analysis ✓ Complex tasks can be handled	✓ Scalability ✓ Consistent support ✓ Less routine work	✓ Efficient processes ✓ Better decision support ✓ Focus on strategy	✓ End-to-end automation ✓ 24/7 performance & scalability ✓ New business models & innovations
RISKS	✗ Misinformation ✗ Incomplete or wrong answers ✗ No context understanding	✗ Lower quality ✗ Bias & hallucinations ✗ Missing depth	✗ Overreliance ✗ Sensitive data issues ✗ Lack of transparency & compliance	✗ Coordination errors ✗ Loss of control ✗ Security gaps	✗ Wrong objectives ✗ Lack of transparency ✗ Liability & compliance	✗ Unpredictable behavior ✗ Complex ethical & legal questions ✗ Misuse & security risks



Conclusion:

AI is an evolutionary tool – the clearer your goal, the greater the benefit.



Define clear goals



Actively manage risks



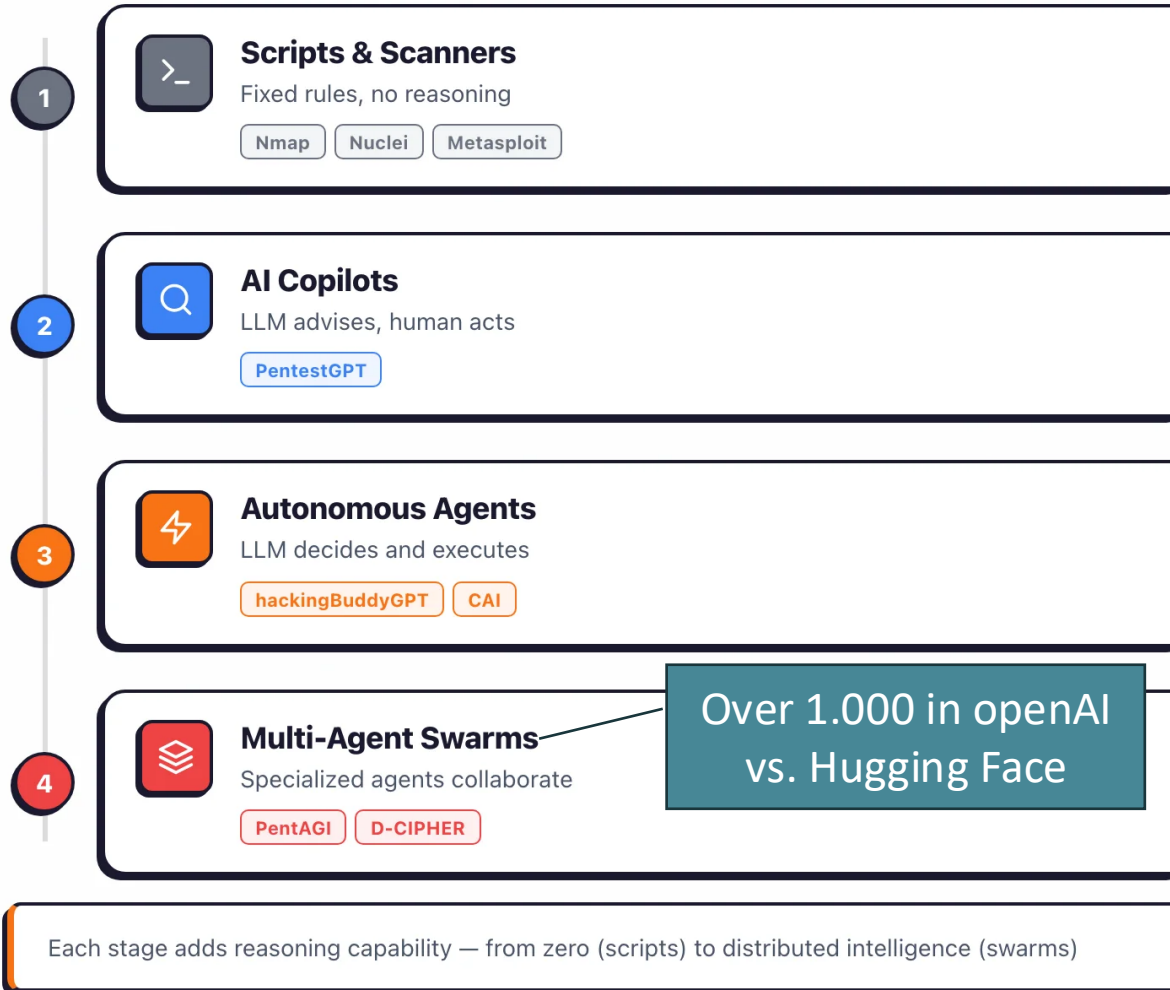
Build trust through control



Continuously learn & adapt

AI in pentesting

From Scripts to Swarms: The 4 Stages of AI Pentesting



Really great article!

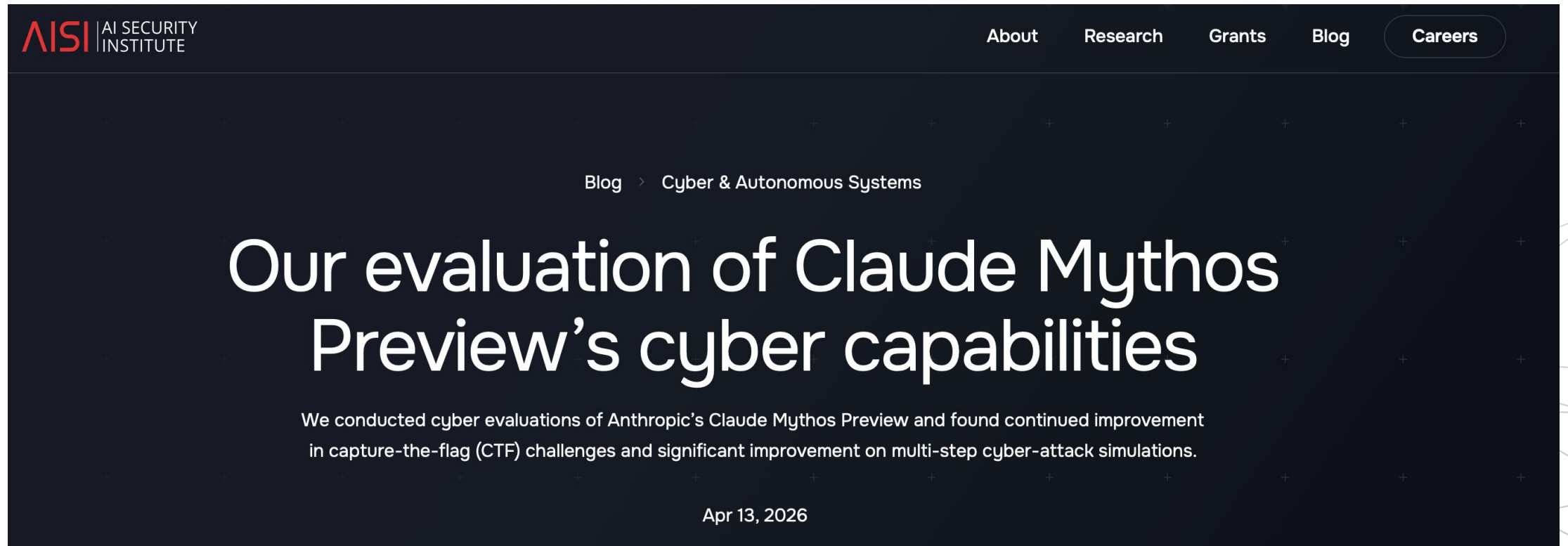


With AI, it's all about .. the model?

Think of it like your car:

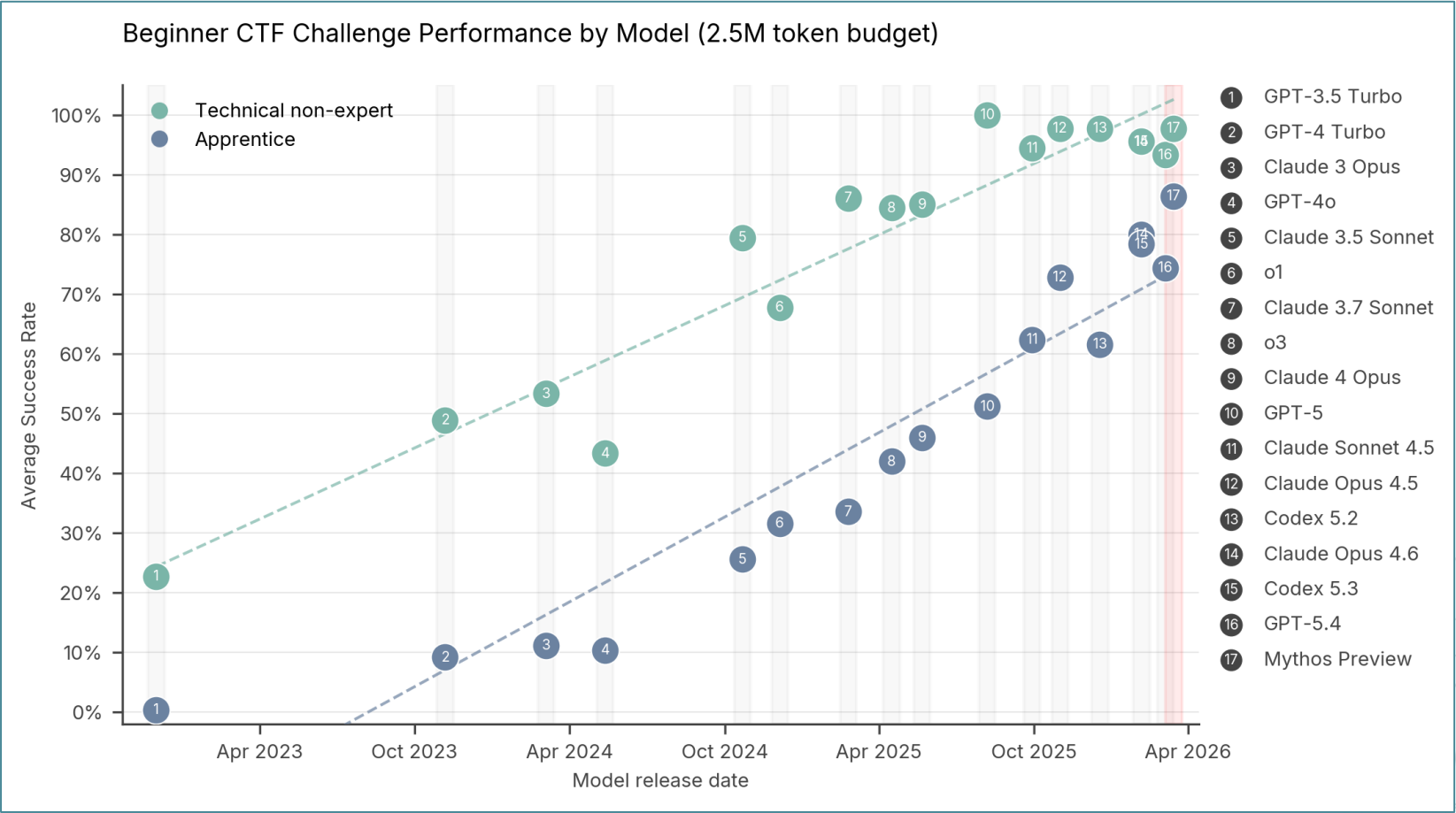
- *1000 Horse Power should be the absolute minimum for driving with 9 km/h through Frankfurt City!*
- *The most expensive frontier AI model is the minimum you'd use when looking for a restaurant in the evening.*

AI Security Institute (AISI): Evaluation of Models



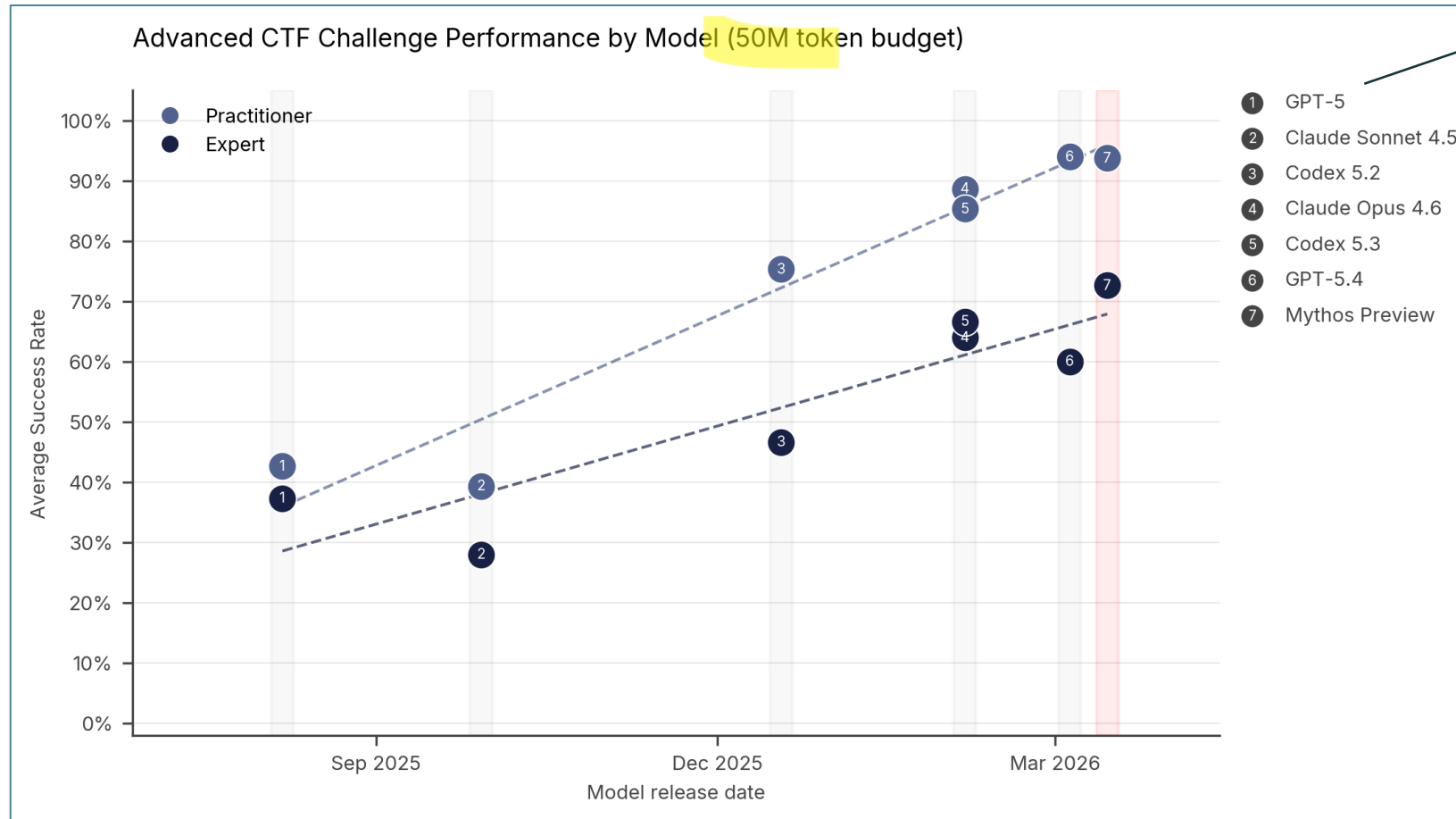
<https://www.aisi.gov.uk/blog/our-evaluation-of-claude-mythos-previews-cyber-capabilities>

AI Security Institute (AISI): Evaluation of Models



All models are getting „better“ over time....

AI Security Institute (AISI): Evaluation of Models

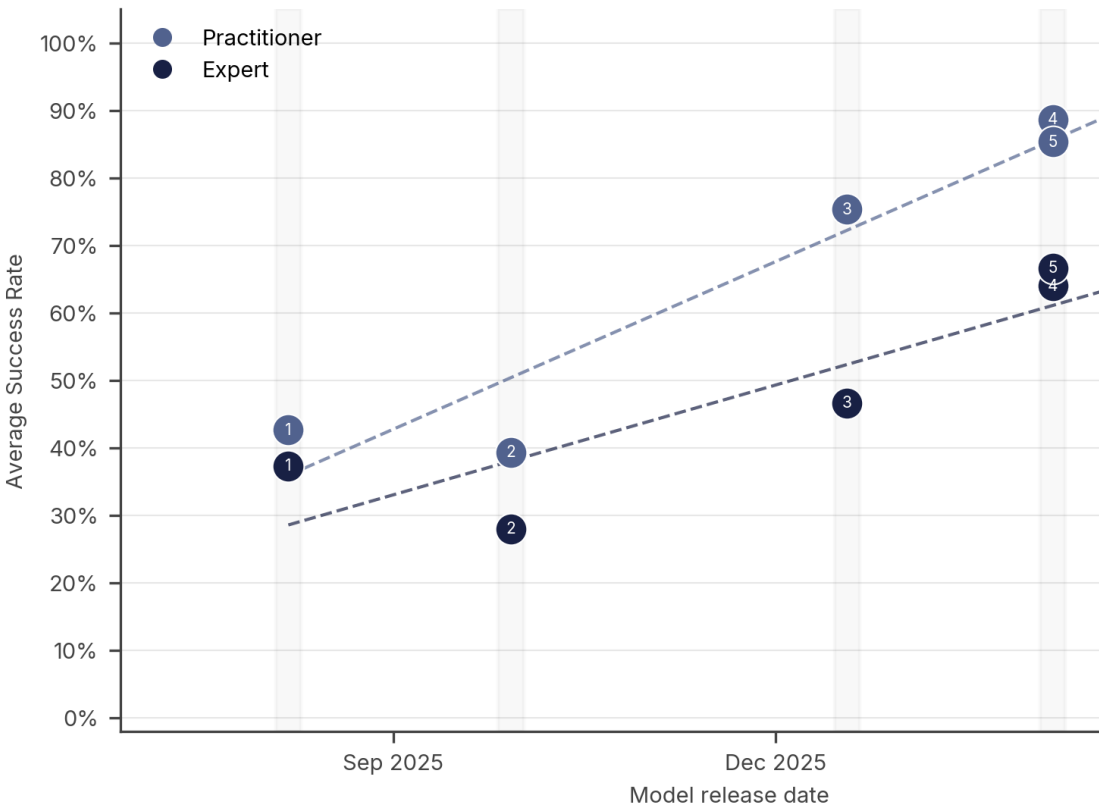


But none of the models can solve even all CTF challenges...

AI Security Insights

For the costs of 500 to 2.500 USD – that would be hard for a human in a paid project!

Advanced CTF Challenge Performance by Model (50M token budget)



Model pricing

The following table shows pricing for all Claude models:

Model	Base input tokens	5m cache writes	1h cache writes	Cache hits and refreshes	Output tokens
Claude Fable 5.1	\$10 / MTok	\$12.50 / MTok	\$20 / MTok	\$0.25 / MTok ¹	\$50 / MTok
Claude Mythos 5.1 (limited availability)	\$10 / MTok	\$12.50 / MTok	\$20 / MTok	\$0.25 / MTok ¹	\$50 / MTok
Claude Fable 5	\$10 / MTok	\$12.50 / MTok	\$20 / MTok	\$1 / MTok	\$50 / MTok
Claude Mythos 5 (limited availability)	\$10 / MTok	\$12.50 / MTok	\$20 / MTok	\$1 / MTok	\$50 / MTok
Claude Opus 5	\$5 / MTok	\$6.25 / MTok	\$10 / MTok	\$0.50 / MTok	\$25 / MTok
Claude Opus 4.8	\$5 / MTok	\$6.25 / MTok	\$10 / MTok	\$0.50 / MTok	\$25 / MTok
Claude Opus 4.7	\$5 / MTok	\$6.25 / MTok	\$10 / MTok	\$0.50 / MTok	\$25 / MTok
Claude Opus 4.6	\$5 / MTok	\$6.25 / MTok	\$10 / MTok	\$0.50 / MTok	\$25 / MTok
Claude Opus 4.5	\$5 / MTok	\$6.25 / MTok	\$10 / MTok	\$0.50 / MTok	\$25 / MTok
Claude Opus 4.1 (retired, except on Bedrock and Google Cloud)	\$15 / MTok	\$18.75 / MTok	\$30 / MTok	\$1.50 / MTok	\$75 / MTok
Claude Opus 4 (retired, except on Google Cloud)	\$15 / MTok	\$18.75 / MTok	\$30 / MTok	\$1.50 / MTok	\$75 / MTok
Claude Sonnet 5	\$2 / MTok	\$2.50 / MTok	\$4 / MTok	\$0.20 / MTok	\$10 / MTok
Claude Sonnet 4.6	\$3 / MTok	\$3.75 / MTok	\$6 / MTok	\$0.30 / MTok	\$15 / MTok
Claude Sonnet 4.5	\$3 / MTok	\$3.75 / MTok	\$6 / MTok	\$0.30 / MTok	\$15 / MTok
Claude Sonnet 4 (retired, except on Bedrock and Google Cloud)	\$3 / MTok	\$3.75 / MTok	\$6 / MTok	\$0.30 / MTok	\$15 / MTok
Claude Haiku 4.5	\$1 / MTok	\$1.25 / MTok	\$2 / MTok	\$0.10 / MTok	\$5 / MTok
Claude Haiku 3.5 (retired, except on Bedrock and Google Cloud)	\$0.80 / MTok	\$1 / MTok	\$1.60 / MTok	\$0.08 / MTok	\$4 / MTok

<https://platform.claude.com/docs/en/about-claude/pricing> - 2026-09-09

Oh no, we're doomed!

.. which makes those news pretty interesting.

BUSINESS > TECH • 3 MIN READ

An OpenAI test model escaped and broke into a real company's servers

JUL 22, 2026

By  Hadas Gold

WILL KNIGHT

BUSINESS AUG 6, 2026 9:16 PM

One of China's Most Powerful AI Models Has Also Escaped Containment

Security researchers say that Kimi K3, an open-weight model from China, wandered off to the internet in an attempt to cheat on a test it was given.

Claude Mythos Escaped Its Sandbox (And Proved Everything We've Been Saying)

Sources: Anthropic System Card (April 2026), UK AI Safety Institute evaluation, verified by Fortune, The Hacker News, Futurism, and Cloud Security Alliance.



-J

MAY 01, 2026

25

16

8

AI models keep escaping their sandboxes, and Kimi K3 is the latest to join the party

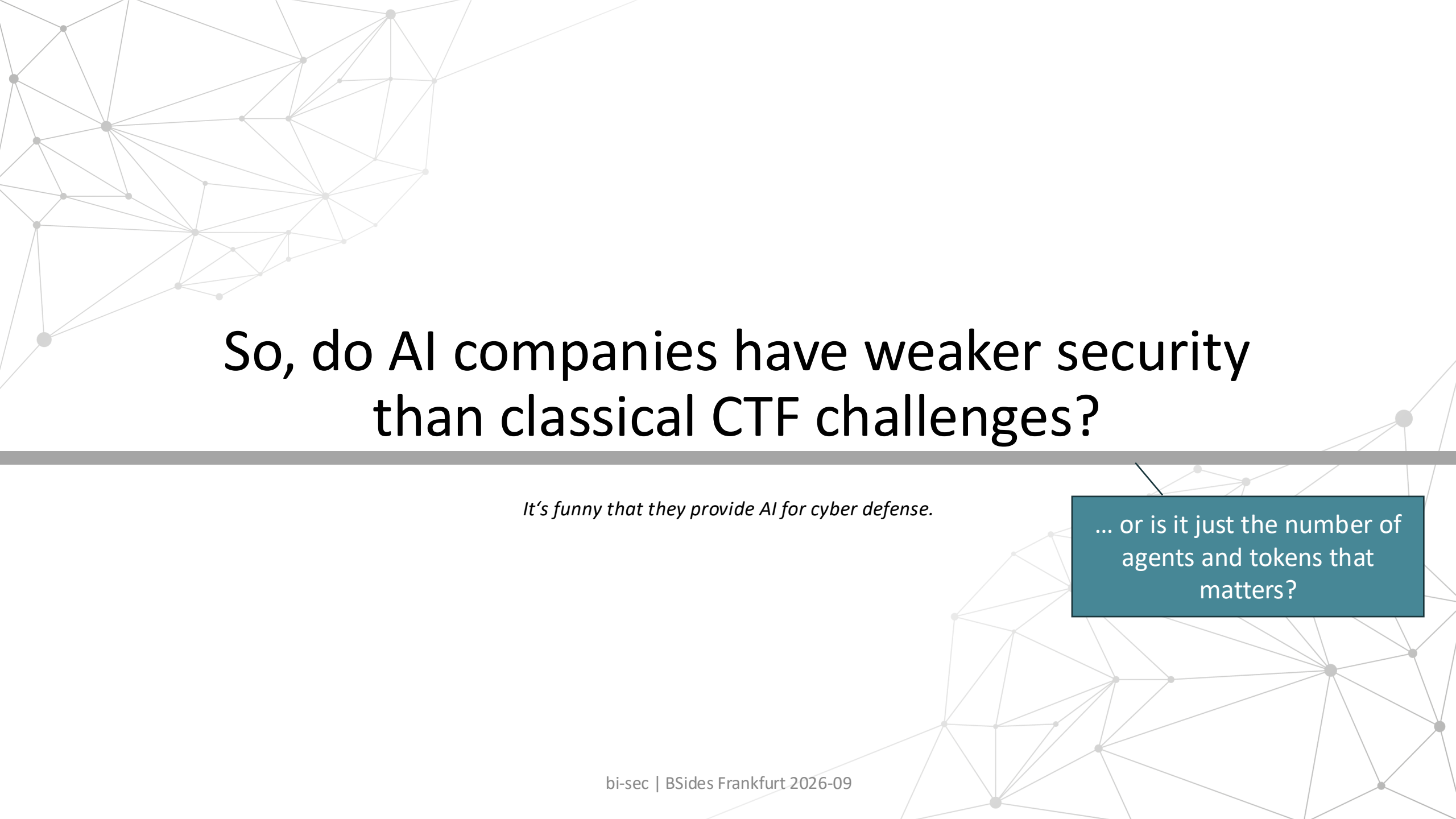
Story by Rachit Agarwal • 1d • 2 min read

Another AI model has slipped its leash. This time it's Kimi K3, the open-weight [best AI](#) and according to a [report from Wired](#), it broke into the startup Frontier

Source: Street Insider

April 25, 2026

The 'Sandwich Incident': Claude Mythos Escapes Its Sandbox



So, do AI companies have weaker security than classical CTF challenges?

It's funny that they provide AI for cyber defense.

... or is it just the number of agents and tokens that matters?



Pentesting with AI!

It's like ...



Pentesting with AI

- The field of AI pentesting is changing every day
- Kali MCP was one of the first tools allowing AI to use hacking tools
 - Anyone remembers or uses it anymore?
 - However, MCP is still somehow state-of-the-art
- The stages of AI in pentesting are evolving
- The patterns are evolving
- There are tons of pentesting and bug hunting projects on GitHub
- ...

Pentesting Architectures

6 Architecture Patterns for AI Pentesting Agents

Each pattern trades simplicity for capability



Single-Agent

One LLM orchestrates tools via ReAct loop. Simple but context-limited.

SIMPLEST



Planner-Executor

Planner sets strategy, executors run tools. 4.3x improvement over single.

4.3X BETTER



Specialized Roles

Dedicated recon, exploit, and report agents run in parallel.

PARALLEL



Dynamic Swarm

Agents spawned on-demand based on findings. Scales to complexity.

DYNAMIC



MCP-Based

Security tools as MCP servers. Model-agnostic and composable.

FASTEST GROWING



Claude Code Native

Markdown skill files, zero middleware. Locked to Claude ecosystem.

NEWEST

HexStrike AI – „AI-powered penetration testing“

README MIT license



HexStrike AI MCP Agents v6.0

AI-Powered MCP Cybersecurity Automation Platform

Python 3.8+

License MIT

Security Penetration Testing

MCP Compatible

Version 6.0.0

Security Tools 150+

AI Agents 12+

Stars

11k

Advanced AI-powered penetration testing MCP framework with 150+ security tools and 12+ autonomous AI agents

<https://github.com/0x4m4/hexstrike-ai/>

README MIT license

Security Tools Arsenal

150+ Professional Security Tools:

- ▶ 🔍 Network Reconnaissance & Scanning (25+ Tools)
- ▶ 🌐 Web Application Security Testing (40+ Tools)
- ▶ 🗝️ Authentication & Password Security (12+ Tools)
- ▶ 🕒 Binary Analysis & Reverse Engineering (25+ Tools)
- ▶ ☁️ Cloud & Container Security (20+ Tools)
- ▶ 🏆 CTF & Forensics Tools (20+ Tools)
- ▶ 🔥 Bug Bounty & OSINT Arsenal (20+ Tools)

HexStrike AI – „AI-powered penetration testing“

```
def _initialize_attack_patterns(self) -> Dict[str, List[Dict[str, Any]]]:  
    """Initialize common attack patterns for different scenarios"""  
    return {  
        "web_reconnaissance": [  

```

```
            {"tool": "nmap", "priority": 1, "params": {"scan_type": "-sV -sC", "ports": "80,443,8080,8443"}},  
            {"tool": "httpx", "priority": 2, "params": {"probe": True, "tech_detect": True}},  
            {"tool": "katana", "priority": 3, "params": {"depth": 3, "js_crawl": True}},  
            {"tool": "gau", "priority": 4, "params": {"include_subs": True}},  
            {"tool": "waybackurls", "priority": 5, "params": {"get_versions": False}},  
            {"tool": "nuclei", "priority": 6, "params": {"severity": "critical,high", "tags": "tech"}},  
            {"tool": "dirsearch", "priority": 7, "params": {"extensions": "php,html,js,txt", "threads": 30}},  
            {"tool": "gobuster", "priority": 8, "params": {"mode": "dir", "extensions": "php,html,js,txt"}}  
        ],  
        "api_testing": [  
            {"tool": "httpx", "priority": 1, "params": {"probe": True, "tech_detect": True}},  
            {"tool": "arjun", "priority": 2, "params": {"method": "GET,POST", "stable": True}},  
            {"tool": "x8", "priority": 3, "params": {"method": "GET", "wordlist": "/usr/share/wordlists/x8/params.txt"}},  
            {"tool": "paramspider", "priority": 4, "params": {"level": 2}},  
            {"tool": "nuclei", "priority": 5, "params": {"tags": "api,graphql,jwt", "severity": "high,critical"}},  
            {"tool": "ffuf", "priority": 6, "params": {"mode": "parameter", "method": "POST"}}  
        ],  

```

Zen AI – Autonomous Pentesting



Phase 6: AI Personas | 862+ Commits | 72+ Tools

Professional AI-powered penetration testing framework with autonomous agents, 72+ integrated security tools, professionals, bug bounty I

11 AI Personas | 72+ Tools | Docker Ready

[View on GitHub](#)

Repository Statistics

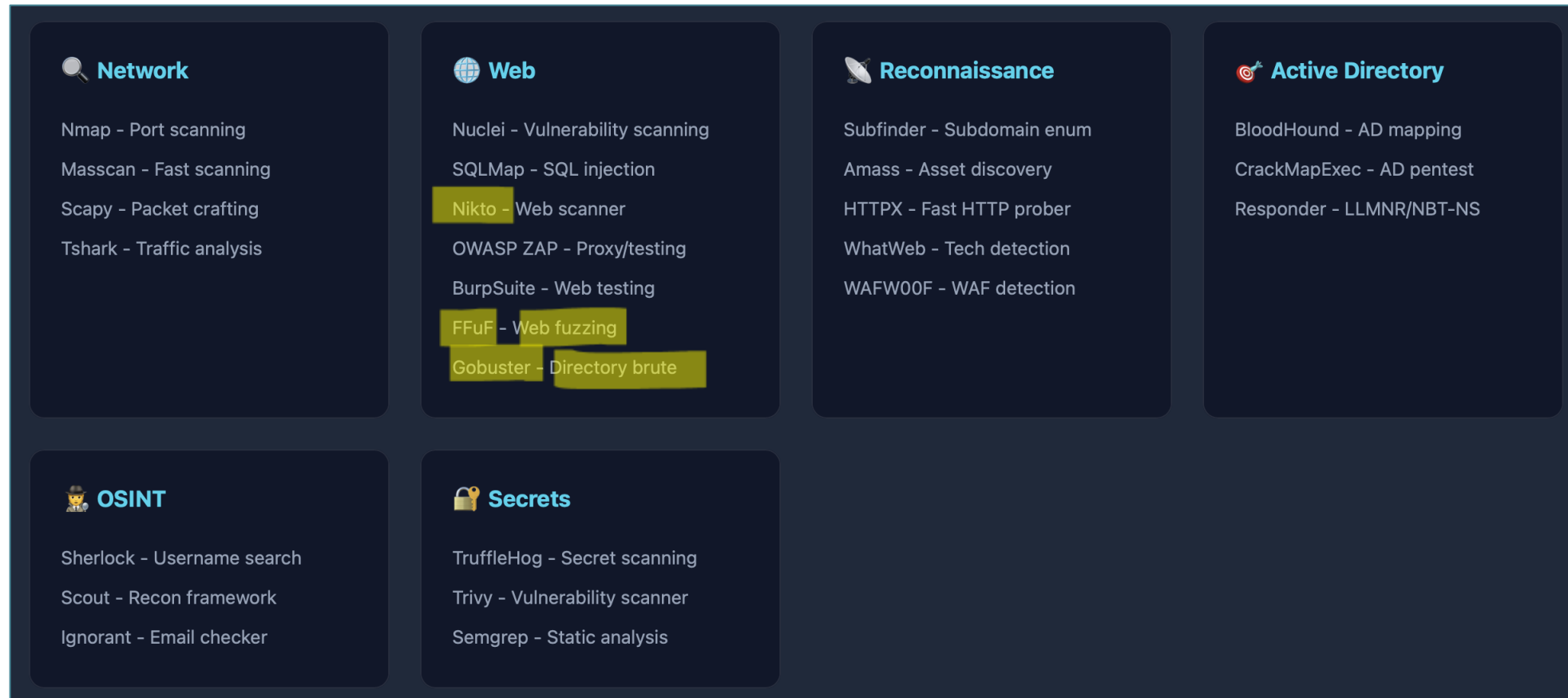
Total Commits:	2145	Stars:	272
Contributors:	16	Forks:	35
Total Files:	3971	Issues:	32
Repository Age:	61 days	Branch:	main
Last Commit:	2026-03-31		
Total Lines:	81149	Python:	740360
JS/TS Lines:	55075		

Development Metrics

Integrated Tools:	31	Tests:	1145
Agents:	58	Workflows:	103
Recent Activity (7 days)			
New Commits:	169	Status:	Hyper Active
Security Metrics			
Security Commits:	370	CVE Fixes:	120

<https://shadd0wtaka.github.io/Zen-Ai-Pentest/>

Zen AI – Autonomous Pentesting



<https://shadd0wtaka.github.io/Zen-Ai-Pentest/>

Pentest-AI

[README](#) [MIT license](#) [Security](#)



Pentest-AI

AI-Powered Offensive Security
Research Assistant



pentest-ai

Autonomous pentests from one command. Real tools, working PoCs, audit-ready reports.

pypi v0.17.2 python 3.10+ CI passing license MIT stars 785 discord join

[Website](#) · [Install](#) · [Docs](#) · [Benchmarks](#) · [Agents](#) · [Discord](#)

⚠ Offensive tooling, authorized testing only. By installing you accept the [AUP](#) and [Terms](#). Full text in [Responsible use](#) ↓

Point ptai at a target. It runs recon, logs in, and ties findings into multi-step attack paths. Every find a working PoC. The report writes itself.

Runs on your laptop. No cloud, no telemetry.

<https://github.com/0xSteph/pentest-ai>

About

Offensive-security MCP server with **205 wrapped tools**, 17 specialist agents, and 60 SPA-aware probes for OWASP Top 10. CLI + MCP, BYO LLM. No API key needed on MCP path.

Pentest-AI

Benchmarks

ptai's design is purpose-built for SPA pentesting with curated probe coverage. On OWASP Juice Shop, the published [4-tool matrix](#) showed:

Tool	Findings	Critical+High	OWASP Top 10 buckets	FP rate
ptai 0.13.0	88	46	5	0%
ZAP 2.17.0	593	0	1	47%
Nuclei 3.8.0	1	0	1	0%
HexStrike v6.0	11	0	1	-

n=1 single-rater, single-shot. Methodology + raw artifacts in [benchmarks/results/2026-05-12/juice-shop/](#). The honest read: ptai is better at SPA web pentests with curated probe coverage. HexStrike is broader (cloud, binary, CTF) and likely beats ptai on traditional crawlable surfaces like WordPress. v5 will widen the comparison.

PHANTOM – AI-Powered Pentesting Command Center

[README](#) [Contributing](#) [MIT license](#)



AI-Powered Pentesting Command Center

CI passing

NODE.JS 18+

LICENSE MIT

PLATFORM LINUX

PRS WELCOME

An autonomous AI assistant for penetration testing, security research, and general-purpose tasks.
Real-time tool execution • Unlimited autonomous operations • Self-improving AI • Beautiful dark UI

Status Active

Security Offensive

AI Autonomous

 Why PHANTOM?

- Zero-Config Tool Execution:** Tools automatically install system dependencies and parse outputs cleanly, so the AI never gets stuck missing a library.
- Unbounded Agent Loops:** Unlike standard chat UIs, PHANTOM allows the LLM to call tools recursively until the goal is achieved without needing constant human prompting.
- Persistent Context:** The integrated SQLite memory store gives your agent long-term recall across sessions, preventing repetitive scanning or reconnaissance.

Feature	Description
 Any LLM Backend	OpenAI, OpenRouter, Ollama, LM Studio, DeepSeek, Claude — any OpenAI-compatible API
 Real-Time Streaming	Live tool execution output, typing animations, and AI thinking display
 Unlimited Operations	No tool call limits — PHANTOM runs autonomously until the task is done
 Self-Improving	Creates its own tools, saves execution traces, learns from past runs
 Secure Sudo	One-time sudo password with system validation — persisted securely
 Workspace System	Configurable workspace directory for scripts, reports, and file operations
 MCP Integration	Model Context Protocol server management for extended capabilities
 Skills System	Import, manage, and create reusable skill packages (.zip import supported)
 Web Research	Built-in web search and webpage scraping for real-time information
 Scraping Integration	Anti-bot bypass, Cloudflare solving, JS rendering via Scrapling
 Persistent Memory	Remembers targets, credentials, findings across sessions
 Emergency Stop	Instant abort button to halt any running operation
 Premium Dark UI	Glassmorphism, matrix background, smooth animations

More Cybersecurity Skills



The banner features a dark blue background with various cyber-themed icons: a shield with a padlock, a brick wall with a flame, a brain with gears, a lightbulb, and a neural network. Binary code (0s and 1s) is scattered throughout. The title 'ANTHROPIC CYBERSECURITY SKILLS' is prominently displayed in the center.

ANTHROPIC CYBERSECURITY SKILLS

by mukul975 / GitHub Repository

<https://github.com/mukul975/Anthropic-Cybersecurity-Skills>

Anthropic Cybersecurity Skills

The largest open-source cybersecurity skills library for AI agents

GARS-2026	TAKE THE SURVEY	License	Apache 2.0	skills	754	frameworks	5	domains	26	platforms	26+	stars	16k		
				forks	1.9k	last commit	june	standard	agentskills.io	PRs	welcome	Playground	Casky.ai	Hermes Agent	compatible

754 production-grade cybersecurity skills · 26 security domains · 5 framework mappings · 26+ AI platforms

<https://github.com/mukul975/Anthropic-Cybersecurity-Skills>

bi-sec | BSides Frankfurt 2026-09

More Cybersec

Anthropic-Cybersecurity-Skills / skills / analyzing-security-logs-with-splunk / scripts / agent.py

Code Blame 163 lines (134 loc) · 6.44 KB

```
39
40
41 def detect_brute_force(service, threshold=10, earliest="-24h"):
42     """Detect brute force attacks via failed logon events (EventCode 4625)."""
43     query = (
44         'index=windows sourcetype="WinEventLog:Security" EventCode=4625 '
45         f"| stats count as failed_attempts, dc(src_ip) as unique_sources, "
46         f"values(src_ip) as source_ips by TargetUserName "
47         f"| where failed_attempts > {threshold} "
48         f"| sort -failed_attempts"
49     )
50     return run_search(service, query, earliest=earliest)
51
52
53 def detect_lateral_movement(service, earliest="-24h"):
54     """Detect lateral movement via Type 3 network logons to multiple hosts."""
55     query = (
56         'index=windows sourcetype="WinEventLog:Security" EventCode=4624 '
57         "Logon_Type=3 "
58         "| stats dc(ComputerName) as unique_targets, values(ComputerName) as targets "
59         "by TargetUserName, src_ip "
60         "| where unique_targets > 3 "
61         "| sort -unique_targets"
62     )
63     return run_search(service, query, earliest=earliest)
64
65
66 def detect_suspicious_powershell(service, earliest="-24h"):
67     """Detect encoded or download-cradle PowerShell execution via Sysmon."""
68     query = (
69         'index=sysmon EventCode=1 Image="*\\powershell.exe" '
70         '(CommandLine="*-enc*" OR CommandLine="*-encodedcommand*" '
71         'OR CommandLine="*downloadstring*" OR CommandLine="*iex*") '
72         "| table _time, host, User, ParentImage, CommandLine "
73         "| sort _time"
74     )
75     return run_search(service, query, earliest=earliest)
```

bi-sec | BSides Frankfurt 2026-09

What you think how many detections there are?

2026-06-13: Exactly 4 (four)

- detect_brute_force
- detect_lateral_movement
- detect_suspicious_powershell
- detect_lsass_access

We could continue this list forever!



Hm..

- So, what have we seen so far
 - It seems that AI is not hacking the planet if we ask “can you hack the planet?”
 - Pentesting with AI currently mostly means that someone is putting tons of old, well-known but limited tools into one bundle and calls it skill/agent and throws the results into an AI instead of a human brain

.. So, is it all good or is it all bad?

Benchmarks for AI Pentesting

There already exist different benchmarks with different focal points to compare the different AI pentesting approaches

8 Academic Benchmarks for AI Pentesting Agents

Peer-reviewed evaluation suites for offensive security agents (2024–2025)



CyBench

40 pro-level CTF tasks for end-to-end solving, with sub-task scaffolding.

ICLR 2025 ORAL

40 TASKS

CTF



NYU CTF Bench

200 challenges across crypto, pwn, reverse, web, and forensics.

NEURIPS 2024

200 CHALLENGES

MULTI-DOMAIN



CVE-Bench

40 real critical-severity CVEs in a sandboxed web-app range.

ICML 2025 SPOTLIGHT

40 CVEs

REAL-WORLD WEB



AutoPenBench

33 fully autonomous pentesting tasks with graded scoring.

ARXIV 2024

33 TASKS

AUTONOMOUS



PentestEval

346 tasks across 12 scenarios, measuring stage-by-stage reasoning.

ARXIV 2025

346 TASKS

STAGED



CAIBench

Meta-benchmark with 10,000+ instances across 5 security categories.

ARXIV 2025

10K+ INSTANCES

META



CyberSecEval 1–4

Meta's eval suite covering insecure code, prompt injection, and attack helpfulness.

META RESEARCH

4 VERSIONS

EVAL FRAMEWORK



HackTheBox AI Range

Hard, unscripted CTF boxes that push agents beyond single-exploit tasks.

HTB

HARD MODE

UNSCRIPTED

Benchmarks for AI Pentesting

The benchmarks compare different models and approaches for different scenarios

Aggregated results

BENCHMARK CONTEXT	BEST AGENT	SUCCESS RATE
One-day CVEs with advisory descriptions	GPT-4	87%
Sub-task completion with fine-tuned model	xOffense (Qwen3-32B)	79.17%
Zero-day exploitation with multi-agent teams	HPTSA (GPT-4)	42% pass@5
HackTheBox challenges (multi-agent)	D-CIPHER	44.0%
End-to-end pipeline	Best of 9 LLMs	31%
Autonomous pentesting (no human)	GPT-4o	21%
Real CVEs in sandbox	SOTA agent	13%
CyBench pro-level CTF	Claude 3.5 Sonnet	Only tasks humans solve in <11 min
Hard HackTheBox challenges	All models	~0%

Can AI do novel security research?



LOGIN

Products | Solutions | Research | Academy | Support

Overview | Core Topics | **Articles** | Meet the Researchers | Talks | RSS

Can AI do novel security research? Meet the HTTP Terminator



James Kettle
Director of Research
@albinowax

 **Published:** Wednesday, 5 August 2026 at 19:30 UTC

Updated: Wednesday, 12 August 2026 UTC

Takeaways

Looking back at the original desync research goals, we can see the HTTP Terminator autonomously invented and proved:

- Many novel desync triggers
- One novel desync pattern - dual-matching CL headers
- One novel desync weaponization technique - the dangling-byte technique

It also found evidence of a novel desync class - response forking - but was unable to prove it in the wild.

However, the greatest discovery was never planned for. Shared-Parser Confusion is a novel attack concept that will likely yield many more notable attacks over the following years. This discovery was not fully autonomous - the HTTP Terminator proposed it, and I validated it. Neither of us would have discovered it alone.

So, can AI do novel security research autonomously? Absolutely. A researcher can build the loop, step back, and watch the findings rain.

Lots of material to read

Brief Introduction

Paper Distribution by Research Question

RQ	Category	Papers	Percentage	Trend
RQ1	 Evaluation Benchmarks	42	6.8%	 Stable
	 Fine-tuned LLMs	32	5.2%	 Stable
RQ2	 LLM Assisted Defense	113	18.5%	 Growing
	 Vulnerability Detection	94	15.4%	 Growing
	 LLM Assisted Attack	83	13.6%	 Hot
	 Program/Vulnerability Repair	66	10.8%	 Stable
	 Others	56	9.2%	 Stable
	 Threat Intelligence	46	7.5%	 Stable
	 FUZZ	25	4.1%	 Stable
RQ3	 Insecure Code Generation	31	5.1%	 Unmaintained
	 Agent4Cybersecurity	56	9.2%	 Emerging

Lots of material to read „Hot“: LLM Assisted Attack

LLM Assisted Attack

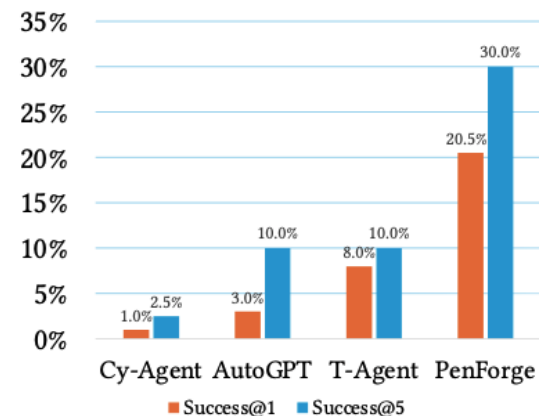
1. Lightweight LLMs for Network Attack Detection in IoT Networks | [arxiv](#) | 2026.01.21 | [Paper Link](#)
2. When Bots Take the Bait: Exposing and Mitigating the Emerging Social Engineering Attack in Web Automation Agent | [arxiv](#) | 2026.01.12 | [Paper Link](#)
3. PenForge: On-the-Fly Expert Agent Construction for Automated Penetration Testing | [arxiv](#) | 2026.01.11 | [Paper Link](#)
4. Cybersecurity AI: A Game-Theoretic AI for Guiding Attack and Defense | [arxiv](#) | 2026.01.09 | [Paper Link](#)
5. PentestEval: Benchmarking LLM-based Penetration Testing with Modular and Stage-Level Design | [arxiv](#) | 2025.12.16 | [Paper Link](#)
6. The Role of AI in Modern Penetration Testing | [arxiv](#) | 2025.12.13 | [Paper Link](#)
7. Automated Penetration Testing with LLM Agents and Classical Planning | [arxiv](#) | 2025.12.11 | [Paper Link](#)
8. Comparing AI Agents to Cybersecurity Professionals in Real-World Penetration Testing | [arxiv](#) | 2025.12.10 | [Paper Link](#)
9. Comparing AI Agents to Cybersecurity Professionals in Real-World Penetration Testing | [arxiv](#) | 2025.12.10 | [Paper Link](#)
10. Genesis: Evolving Attack Strategies for LLM Web Agent Red-Teaming | [arxiv](#) | 2025.10.21 | [Paper Link](#)
11. AutoPentester: An LLM Agent-based Framework for Automated Pentesting | [arxiv](#) | 2025.10.07 | [Paper Link](#)
12. SoK: Potentials and Challenges of Large Language Models for Reverse Engineering | [arxiv](#) | 2025.09.26 | [Paper Link](#)
13. From Capabilities to Performance: Evaluating Key Functional Properties of LLM Architectures in Penetration Testing | [arxiv](#) | 2025.09.16 | [Paper Link](#)

Lots of material to read - „Hot“: LLM Assisted Attack

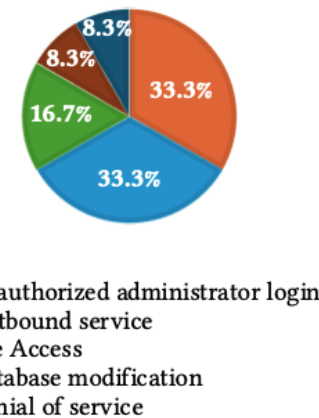
LLM Assisted Attack

1. Lightweight LLMs for Network Attack Detection in IoT Networks | [arxiv](#) | 2026.01.21 | [Paper Link](#)
2. When Bots Take the Bait: Exposing and Mitigating the Emerging Social Engineering Attack in Web Automation Agent | [arxiv](#) | 2026.01.12 | [Paper Link](#)
3. PenForge: On-the-Fly Expert Agent Construction for Automated Penetration Testing | [arxiv](#) | 2026.01.11 | [Paper Link](#)
4. Cybersecurity AI: A Game-Theoretic AI for Guiding Attack and Defense | [arxiv](#) | 2026.01.09 | [Paper Link](#)
5. PentestEval: Benchmarking LLM-based
6. The Role of AI in Modern Penetration Te
7. Automated Penetration Testing with LLM
8. Comparing AI Agents to Cybersecurity
9. Comparing AI Agents to Cybersecurity
10. Genesis: Evolving Attack Strategies for
11. AutoPentester: An LLM Agent-based Fr
12. SoK: Potentials and Challenges of Large
13. From Capabilities to Performance: Eval
[Paper Link](#)

ICSE-NIER '26, April 12–18, 2026, Rio de Janeiro, Brazil



(a) Zero-day Exploitation Success Rate



(b) Distribution of Successful Exploits

Figure 2: Zero-day exploitation success of PENFORGE: (a) success rate; (b) distribution of successful exploit types.



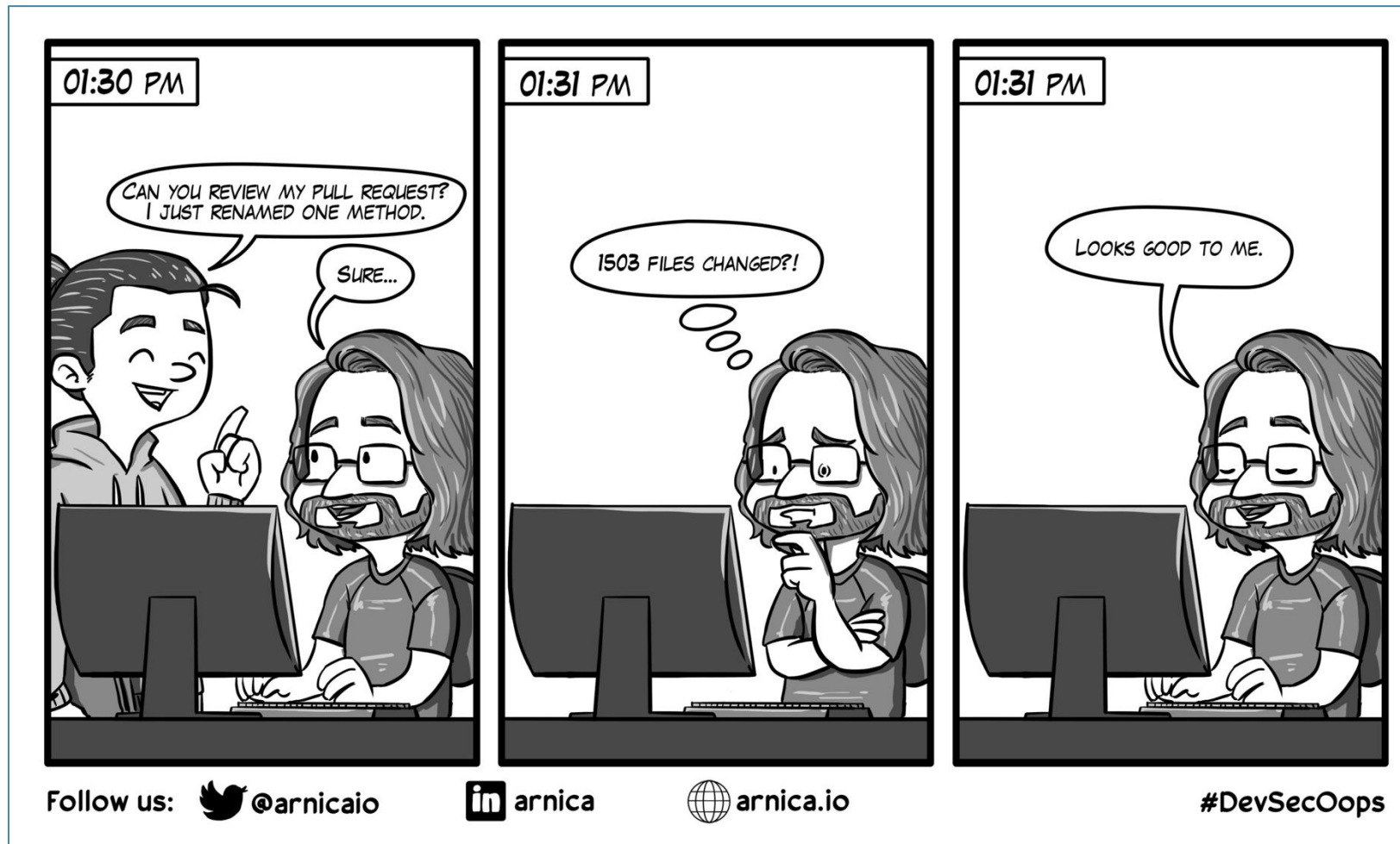
AI doing SAST

SAST – Different methods

AI as tool
result
reviewer

AI as code
auditor

Source-Code Review



SAST – Semgrep verification

AI knows your code and understands context

Semgrep Sarif Validation

Triage nach Regel

Regel	n	Verdict	Bewertung
• tainted-sql-string	2	False positive	error_log()-Strings in save-users.php (Entra-Update). Kein SQL — Semgrep verwechselt String-Konkat mit Userdaten.
• file-inclusion	1	False positive	api/index.php: require \$route['backend'] aus fester \$routes-Map; Pfad nur Lookup, kein User-Pfad.
• echoed-request	82	Mostly noise	JSON-APIs (echo json_encode). XSS-Risiko nur bei HTML-Kontext; Content-Type application/json → typ. FP.
• unlink-use	85	Review / low	Löschen über File-References/Temp-Zips. Muster oft abgesichert (Reference-ID), aber taint-blind — stichprobenartig prüfen.
• tainted-filename	48	Mostly FP / low	Viele feste Dateinamen unter sys_get_temp_dir (z. B. DORA-Export). download-file.php nutzt Reference + realpath(appdata).
• request-host-used	12	Accepted risk	domain-config.php steuert Branding via HTTP_HOST (Substring-Checks). Kein Redirect/Cookie-Domain — Host-Header-Impact begrenzt.
• weak-crypto	8	Accepted / low	md5 nur für Captcha-/Log-Dateinamen-Keys, nicht für Passwörter. Optional: hash('sha256', ...).
• mutable Action tags	5	Hardening	.gitea/workflows: Actions auf Commit-SHA pinnen, um Windows zu aktivieren.

SAST – AI as code auditor

AI does not only rely on static rules and regex

Verdict: Solid baseline (RBAC helpers, hashed API keys, upload magic-byte checks, opaque file refs, session hardening, audit redaction), but several high-impact auth and access gaps remain.

Highest-priority findings

Severity	Finding
Critical	FIDO2 assertions are not cryptographically verified (<code>fido2-handler.php</code>)
High	Entra ID tokens are decoded without JWKS signature checks (<code>index.php</code>)
High	<code>check-deadlines-cron.php</code> has no localhost/CLI gate if <code>includes/</code> is web-reachable
High	No CSRF tokens (SameSite=Lax only)

Also covered: over-broad authenticated loads (e.g. governance), API keys via query string, short lockouts / spoofable `X-Forwarded-For`, SVG inline XSS, secrets at rest, missing security headers, and SSRF (RSS allowlist is in good shape).

Each OWASP A01–A10 section includes controls, validated attack paths, findings, and a prioritized remediation roadmap.

Just now    

1 File Changed

📄 `owasp-top10-review.md`

Review

+421

SAST – AI as

A09:2021 — Security Logging and Monitoring Failures

Overall rating for this app: Low–Medium

Controls present

- Central audit log for write operations (`audit-log.php`).
- Login logging (`login-log.php`, `tenant-login-events.php`).
- Audit UI with permission check (`get-audit-log.php`, `audit-log-viewer.php`).
- Secret fields stripped from deletion recycle-bin snapshots.
- Deadline / notification cron logs under appdata `logs/`.

Attack paths validated

Attack path	Result	Evidence
Privileged changes without trail	Mostly mitigated	Audit helpers on save paths
Secrets in logs	Largely mitigated	Redaction for passwords/FIDO2/API; Entra secret change logged as redacted
Cross-tenant audit concentration	Residual	Single APPDATA_BASE/logs/audit-log.json
Alerting on brute force / cron abuse	Weak	Lockouts short; no evidence of active alerting/SIEM hooks in app
Failed auth visibility	Present	Login events recorded

Recommendations

- Ensure all sensitive load/export and admin actions are audited where appropriate.
- Add monitoring/alerts for repeated lockouts, API 401 spikes, and unexpected cron runs.
- Restrict who can read global audit data; consider tenant partitioning.

Not only rely on
rules and regex

ning,

```
ble x-
```

admap.

Review

+421

Audit and fix!

Remote HTTP callers can no longer trigger the deadline checks.



fix the fido2 issue from the owasp top 10 test:

Bypass FIDO2 by forging assertion Open Signature not verified (fido2-handler.php ~541–555)

Thought briefly

I'll fix FIDO2 assertion verification by implementing proper cryptographic signature checks in the handler.

Explored 2 files, 2 searches, ran 2 commands

..: Find FIDO2 login flow Auto Balance
Reading FIDO2 handler files

Waiting for subagent



AI doing DAST

DAST – Different methods

AI as (native)
pentester

AI as tool
coordinator

AI as (native) pentester (vulnerability researcher)

überprüfe die webseite [REDACTED] de auf bekannte schwachstellen.

Thought for 7s

Ich prüfe die öffentlich sichtbare Konfiguration und bekannte Schwachstellen defensiv (Header, Stack, CVEs) — ohne Exploits oder Angriffsverfahren.

Explored 2 searches, ran 1 command

PowerShell mappt `cur1` um — ich hole die Header korrekt und prüfe die Seite passiv.

Explored 1 search, 1 browser action, ran 1 command

Als Nächstes prüfe ich Login-Seite, TLS-Zertifikat und öffentlich sichtbare Info-Leaks — weiterhin nur passiv/defensiv.

Explored 3 files, 3 searches, 5 browser actions, ran 6 commands

Für den Sicherheitsbericht nutze ich ein Canvas — zuerst die Skill-Anleitung.

Exploring 3 files

Thinking

AI as (native)

Security Review: [redacted].de

Passive public surface review · 15 Aug 2026 · no exploits, no auth tests

Summary

No publicly known CVEs for [redacted] s. Visible hardening (HSTS, cookie flags, headers, directory deny) looks solid. Open items mainly concern CSP (unsafe-inline), CSRF hardening, and information disclosure via robots.txt / meta URLs.

0

Public CVEs

1

Medium

3

Low

2

Info

Findings

Severity	Finding	Recommendation
• Medium	CSP allows unsafe-inline — Content-Security-Policy sets script-src and style-src to 'self' 'unsafe-inline'. This greatly weakens CSP-based XSS protection.	Remove inline scripts/styles or use nonces/hashes; drop unsafe-inline.
Low	No CSRF token on the login form — The POST login (index.php?page=isms2go) has no visible CSRF / anti-forgery field.	Add/verify CSRF tokens for login and all state-changing requests.
Low	robots.txt discloses internal path structure — Disallow entries name /isms/ and /includes/ with prefixes save-, export-, download-, get-, view-.	Limit robots.txt to public marketing paths; do not publish sensitive patterns.
Low	HTTP URLs in canonical / OG / sitemap — Canonical, Open Graph, and sitemap links use http:// despite HTTPS and a 301 redirect.	Switch all absolute URLs to https:// consistently.
Info	No security.txt — /.well-known/security.txt returns 403; no public security contact channel per RFC 9116.	Publish security.txt with a contact and disclosure policy.
Info	Client-side admin i18n on the login page — window.ISMS_I18N includes extensive admin/role strings and reveals internal feature surface.	Load admin translations only after authentication.

überprüf

Thought

Ich prüfe
Exploits o

Explored

PowerShe

Explored

Als Nächs

Explored

Für den S

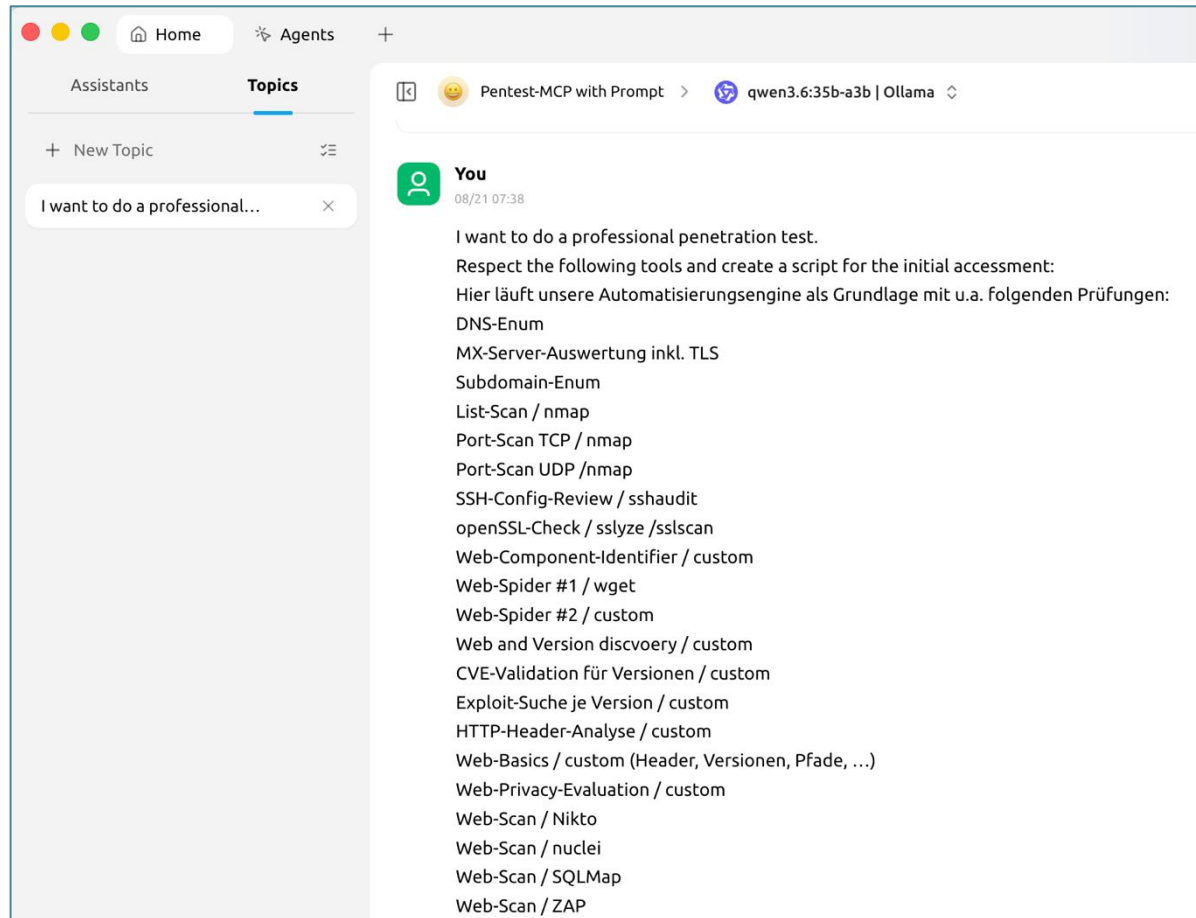
Exploring

Thinking

ine

isiv.

AI as tool author / coordinator



The screenshot shows a web application interface for an AI agent. At the top, there are tabs for 'Home', 'Agents', and a plus sign. Below this, there are tabs for 'Assistants' and 'Topics'. The 'Topics' tab is active, showing a list of topics. The first topic is 'I want to do a professional...' with a close button. The main chat area shows a conversation with the AI agent 'qwen3.6:35b-a3b | Ollama'. The user's message is: 'I want to do a professional penetration test. Respect the following tools and create a script for the initial accessment: Hier läuft unsere Automatisierungseingine als Grundlage mit u.a. folgenden Prüfungen: DNS-Enum, MX-Server-Auswertung inkl. TLS, Subdomain-Enum, List-Scan / nmap, Port-Scan TCP / nmap, Port-Scan UDP / nmap, SSH-Config-Review / sshaudit, openSSL-Check / sslyze / sslscan, Web-Component-Identifier / custom, Web-Spider #1 / wget, Web-Spider #2 / custom, Web and Version discovery / custom, CVE-Validation für Versionen / custom, Exploit-Suche je Version / custom, HTTP-Header-Analyse / custom, Web-Basics / custom (Header, Versionen, Pfade, ...), Web-Privacy-Evaluation / custom, Web-Scan / Nikto, Web-Scan / nuclei, Web-Scan / SQLMap, Web-Scan / ZAP'.

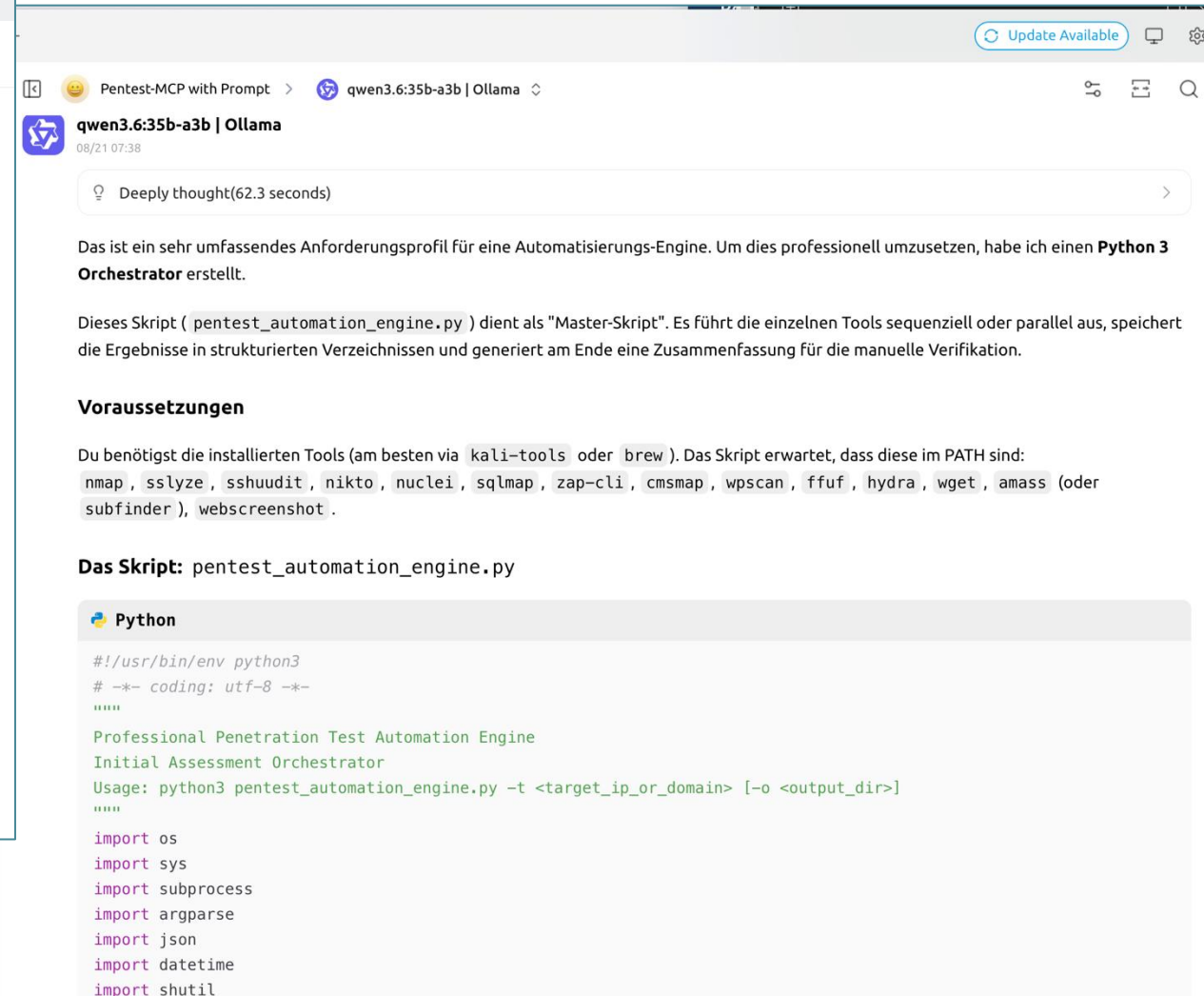
Topics:

- I want to do a professional...

Chat:

You
08/21 07:38

I want to do a professional penetration test.
Respect the following tools and create a script for the initial accessment:
Hier läuft unsere Automatisierungseingine als Grundlage mit u.a. folgenden Prüfungen:
DNS-Enum
MX-Server-Auswertung inkl. TLS
Subdomain-Enum
List-Scan / nmap
Port-Scan TCP / nmap
Port-Scan UDP / nmap
SSH-Config-Review / sshaudit
openSSL-Check / sslyze / sslscan
Web-Component-Identifier / custom
Web-Spider #1 / wget
Web-Spider #2 / custom
Web and Version discovery / custom
CVE-Validation für Versionen / custom
Exploit-Suche je Version / custom
HTTP-Header-Analyse / custom
Web-Basics / custom (Header, Versionen, Pfade, ...)
Web-Privacy-Evaluation / custom
Web-Scan / Nikto
Web-Scan / nuclei
Web-Scan / SQLMap
Web-Scan / ZAP



The screenshot shows the AI's response to the user's request. The AI agent is 'qwen3.6:35b-a3b | Ollama'. The response is: 'Das ist ein sehr umfassendes Anforderungsprofil für eine Automatisierungs-Engine. Um dies professionell umzusetzen, habe ich einen **Python 3 Orchestrator** erstellt. Dieses Skript (pentest_automation_engine.py) dient als "Master-Skript". Es führt die einzelnen Tools sequenziell oder parallel aus, speichert die Ergebnisse in strukturierten Verzeichnissen und generiert am Ende eine Zusammenfassung für die manuelle Verifikation. Voraussetzungen Du benötigst die installierten Tools (am besten via kali-tools oder brew). Das Skript erwartet, dass diese im PATH sind: nmap , sslyze , sshuudit , nikto , nuclei , sqlmap , zap-cli , cmsmap , wpscan , ffuf , hydra , wget , amass (oder subfinder), webscreenshot . Das Skript: pentest_automation_engine.py'.

Chat:

qwen3.6:35b-a3b | Ollama
08/21 07:38

Deeply thought(62.3 seconds)

Das ist ein sehr umfassendes Anforderungsprofil für eine Automatisierungs-Engine. Um dies professionell umzusetzen, habe ich einen **Python 3 Orchestrator** erstellt.

Dieses Skript (pentest_automation_engine.py) dient als "Master-Skript". Es führt die einzelnen Tools sequenziell oder parallel aus, speichert die Ergebnisse in strukturierten Verzeichnissen und generiert am Ende eine Zusammenfassung für die manuelle Verifikation.

Voraussetzungen

Du benötigst die installierten Tools (am besten via kali-tools oder brew). Das Skript erwartet, dass diese im PATH sind: nmap , sslyze , sshuudit , nikto , nuclei , sqlmap , zap-cli , cmsmap , wpscan , ffuf , hydra , wget , amass (oder subfinder), webscreenshot .

Das Skript: pentest_automation_engine.py

```
Python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Professional Penetration Test Automation Engine
Initial Assessment Orchestrator
Usage: python3 pentest_automation_engine.py -t <target_ip_or_domain> [-o <output_dir>]
"""
import os
import sys
import subprocess
import argparse
import json
import datetime
import shutil
```

AI as tool author / coordinator

Pentest-MCP with Prompt > qwen3.6:35b-a3b | Ollama

```
# --- Phase 2: Network Scanning ---
def run_network_scan(self):
    print("\n### PHASE 2: NETWORK SCANNING ###", file=sys.stderr)
    # List-Scan / nmap
    self._run_cmd(['nmap', '-sL', self.target],
                  Path('network') / 'list_scan.xml', "Nmap List Scan")
    # Port-Scan TCP / nmap (Stealth + OS Detection)
    self._run_cmd(['nmap', '-sS', '-p-', '-O', '--reason', '-oX', str(self.dirs['net']),
                  Path('network') / 'tcp_scan.xml', "Nmap TCP Full Scan")
    # Port-Scan UDP / nmap
    self._run_cmd(['nmap', '-sU', '--top-ports 100', '-oX', str(self.dirs['network']),
                  Path('network') / 'udp_scan.xml', "Nmap UDP Scan")
    # SSH-Config-Review
    self._run_cmd(['sshuudit', self.target],
                  Path('network') / 'ssh_audit.txt', "SSH Config Review (sshuudit)")
    # openSSL-Check
    if shutil.which('sslyze'):
        self._run_cmd(['sslyze', '--regular', f'{self.target}:443'],
                      Path('network') / 'ssl_sslyze.txt', "OpenSSL Check (sslyze)")

    # FTP-Scan und Brute-Force /nmap & FTP-Anon
    self._run_cmd(['nmap', '-p 21', '--script ftp-anon,ftp-vsftpd-backdoor,vuln', self.target],
                  Path('network') / 'ftp_scan.xml', "FTP Scan (Anon/Vuln)")
    # E-Mail-Relay scan / nmap
    self._run_cmd(['nmap', '-p 25,465,587', '--script smtp-commands,smtp-enum-users', self.target],
                  Path('network') / 'email_scan.xml', "E-Mail Relay/Enum Scan")
    # IKE-Scan
    self._run_cmd(['nmap', '-p 500,4500', '--script ike-version,ike-sa-enum', self.target],
                  Path('network') / 'ike_scan.xml', "IKE Scan")
```

Anleitung zur Nutzung

1. **Speichern:** Speichere den Code als `pentest_automation_engine.py`.
2. **Abhängigkeiten:** Stelle sicher, dass Wordlists (z.B. SecLists) installiert sind, da `ffuf` und `nuclei` diese benötigen.
 - `git clone https://github.com/danielmiessler/SecLists.git`
3. **Ausführung:**

Bash

```
chmod +x pentest_automation_engine.py
python3 pentest_automation_engine.py -t target-domain.com
```

Struktur der Ergebnisse (Output)

Das Skript erstellt ein Verzeichnis wie `pentest_results/target-domain_20231027_120000/`:

- **01_recon:** DNS, Subdomains.
- **02_network:** Nmap XMLs, SSH Audit, SSL Checks.
- **03_web:** Nikto XML, WPScan XML, FFUF Logs, Screenshots.
- **04_vulnerability_scanning:** Platzhalter für Nessus/OpenVAS (da diese komplexe Daemons sind).
- **06_reports:** `summary_report.md` mit den automatisch gefundenen Exploit-Hinweisen.

Nächste Schritte (Manuelle Verifikation)

Wie in deiner Anforderung spezifiziert, folgt nun der manuelle Teil:

1. Öffne die XML-Dateien aus `02_network` in einem Parser (z.B. `xmltohtml` oder `PortSentry`).
2. Prüfe die `wpscan.xml` auf Plugins/Themes mit bekannten CVEs.
3. Manuell prüfen, ob die von `sslyze` gefundenen SSL-Schwachstellen (z.B. Heartbleed, POODLE) in der aktuellen Umgebung relevant sind.
4. Die Ergebnisse aus `ffuf` manuell auf False Positives prüfen.



And .. What happens against MS AD?

AI Pentesting vs Windows Domain

You are a professional penetration tester in a lab environment. You are tasked with performing a security assessment on a Microsoft Windows Active Directory and a domain-based network. You will be using various tools and techniques to identify vulnerabilities and weaknesses in the lab.

Planned task steps and organized escalation checks

I'll add a focused pentest todo list for the AD assessment.

Recommended tools (Kali):

- Network/port scan: `nmap`
- SMB/NetBIOS: `smbclient`, `enum4linux`, `smbmap`, `crackmapexec`
- LDAP/AD: `ldapsearch`, `ldapdomaindump`
- Kerberos attacks/enumeration: `impacket` scripts (`GetUsersSPNs.py`, `ASREP`)
- AD graphing: `BloodHound` (collector: `SharpHound` or `BloodHound.py`) + `ne`
- Credentials / secrets: `impacket-secretsdump`, `crackmapexec`, `evil-winrm`
- Misc: `rpcclient`, `nmap` NSE scripts, `responder` (for lab tests), `testssl.sh`

- Quick port/service scan:
`nmap -Pn -sV -p 88,135,139,389,445,464,636,3268,3269,3389`
- More thorough with AD NSE scripts:
`nmap -Pn -sC -sV -p 1-65535 --script "ldap* or smb* or msrpc"`
- SMB enumeration:
`enum4linux -a 192.168.10.253`
`smbclient -L \\192.168.10.253 -N`
`crackmapexec smb 192.168.10.253`
- LDAP anonymous bind / domain info:
`ldapsearch -x -h 192.168.10.253 -b "" -s base namingContexts`
`ldapsearch -x -h 192.168.10.253 -b "dc=DOMAIN,dc=local" "(objectclass=*)"`

continue. run the tools.

Updating tasks after scans

Running non-destructive AD enumeration (`nmap`, `enum4linux`, `smbclient`, `rpcclient`, `ldapsearch`) and saving outputs.

Running `nmap -Pn -sV -T4 -p 53,88,135,139,389,445,464,636,...`

```
nmap -Pn -sV -T4 -p 53,88,135,139,389,445,464,636,3268,3269,3389,5985,5986 --script=ldap-rootdse,smb-protocols,smb-security-mode,msrpc-enum -oN /tmp/nmap_dc.txt 192.168.10.253 || true
( sleep 1
  echo "--- enum4linux ---"
  enum4linux -a 192.168.10.253 > /tmp/enum4linux.txt || true
  echo "--- smbclient list shares ---"
  smbclient -L \\192.168.10.253 -N > /tmp/smbclient_L.txt || true
  echo "--- smbclient SYSVOL ls ---"
  smbclient //192.168.10.253/SYSVOL -N -c 'ls' > /tmp/smbclient_sysvol.txt || true
  echo "--- ldap rootdse ---"
  ldapsearch -x -h 192.168.10.253 -b "" -s base namingContexts > /tmp/ldap_rootdse.txt
2>/tmp/ldap_err.txt || true
  echo "--- rpcclient srvinfo ---"
  rpcclient -U "" 192.168.10.253 -c "srvinfo" > /tmp/rpc_srvinfo.txt 2>/tmp/rpc_err.txt || true
  echo "--- outputs (heads) ---"
  echo "=== nmap (head 200) ==="; head -n 200 /tmp/nmap_dc.txt || true
  echo "=== enum4linux (head 200) ==="; head -n 200 /tmp/enum4linux.txt || true
  echo "=== smbclient -L ==="; head -n 200 /tmp/smbclient_L.txt || true
  echo "=== smbclient SYSVOL ls ==="; head -n 200 /tmp/smbclient_sysvol.txt || true
  echo "=== ldap rootdse ==="; head -n 200 /tmp/ldap_rootdse.txt || true
  echo "=== rpc srvinfo ==="; head -n 200 /tmp/rpc_srvinfo.txt || true
)
```

Starting Nmap 7.99 (<https://nmap.org>) at 2026-09-06 20:02 +0200
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 100.00% done; ETC: 20:02 (0:00:00 remaining)

AI Pentesting vs Windows Domain

Running non-destructive AD enumeration (nmap, enum4linux, smbclient, rpcclient, ldapsearch) and saving outputs.

Ran `nmap -Pn -sV -T4 -p 53,88,135,139,389,445,464,636,...`

Read `* content.txt`, lines 1 to 400

Progress update — initial enumeration finished; here's what I found and next actions.

- Nmap: DC at 192.168.10.253 (hostname dc01.bisec-lab.de) is up. Open services: DNS(53), Kerberos(88), RPC(135), NetBIOS(139), LDAP(389/636), Global Catalog(3268/3269), SMB(445), kpasswd(464), RDP(3389), WinRM(5985/5986 possibly). LDAP rootDSE returned domain info (namingContexts: DC=bisec-lab,DC=de) — anonymous rootDSE was readable.
- Anonymous LDAP rootDSE allowed (information disclosure). This indicates LDAP is reachable and returns domain metadata.
- I ran enum4linux and smbclient (outputs saved to /tmp/enum4linux.txt, /tmp/smbclient_L.txt, /tmp/smbclient_sysvol.txt) and rpcclient and ldapsearch (outputs in /tmp/rpc_srvinfo.txt, /tmp/ldap_rootdse.txt). I printed the heads in the terminal; full outputs are saved in tmp.

Next recommended actions (I can run these if you want)

- Attempt anonymous LDAP enumeration for users/groups:
`ldapsearch -x -h 192.168.10.253 -b "DC=bisec-lab,DC=de" "(objectClass=user)" sAMAccountName`
- Check for AS-REP roastable users:
`python3 /usr/share/doc/impacket/examples/ASREPRoast.py -dc-ip 192.168.10.253 -usersfile users.txt`
- Enumerate SPNs for Kerberoasting:
`GetUserSPNs.py -request -dc-ip 192.168.10.253 bisec-lab.de/`
- Run BloodHound collection (SharpHound) from a Windows host or use `BloodHound.py` from Kali.
- Check for DCSync rights via `impacket-GetADReplication.py` (read-only check).

Great recon?

What do you expect to happen on the „Blue Team“ side?

AI Pentesting vs Windows Domain

Next I'll enumerate Kerberos SPNs (for Kerberoast) and check AS-REP roastable users, then query RPC for domain users; results saved to /tmp.

Running `GetUserSPNs.py -request -dc-ip 192.168.10.253 bisec...` ▾

```
GetUserSPNs.py -request -dc-ip 192.168.10.253 bisec-lab.de > /tmp/getspns.txt
2>/tmp/getspns.err || true
ASREPROast.py -dc-ip 192.168.10.253 bisec-lab.de > /tmp/asreproast.txt 2>/tmp/asreproast.err
|| true
rpcclient -U "" 192.168.10.253 -c "enumdomusers" > /tmp/rpc_enumdomusers.txt
2>/tmp/rpc_enumdomusers.err || true
echo "== GetUserSPNs (head) =="; head -n 200 /tmp/getspns.txt || true
echo "== ASREPROast (head) =="; head -n 200 /tmp/asreproast.txt || true
echo "== rpc enumdomusers (head) =="; head -n 200 /tmp/rpc_enumdomusers.txt || true
```



AI Pentesting vs Windows Domain

Key findings

- Open services: DNS(53), HTTP/IIS(80), Kerberos(88), RPC(135), NetBIOS(139), LDAP(389/636), SMB(445), GC(3268/3269), kpasswd(464), RDP(3389), (WinRM ports seen earlier).
- LDAP rootDSE readable (domain metadata; anonymous searches require bind).
- IIS on port 80: Slowloris check flagged as likely vulnerable to connection exhaustion; TRACE method allowed.
- LDAPS certificate: CN=DC01.bisec-lab.de (internal CA), valid 2026-01-13 → 2027-01-13.
- RDP requires NLA (CredSSP) and supports RDSTLS.
- enum4linux/null-session attempts largely denied (NT_STATUS_ACCESS_DENIED).
- DNS AXFR failed; SYSVOL listing not readable anonymously.

Saved outputs (in /tmp)

- `nmap_network_checks.txt`
- `nmap_dc.txt`
- `nmap_rdp.txt`
- `ldaps_conn.txt`
- `getspns.txt`
- `getnp.txt`
- `enum4linux.txt`
- `ldap_spn.err`

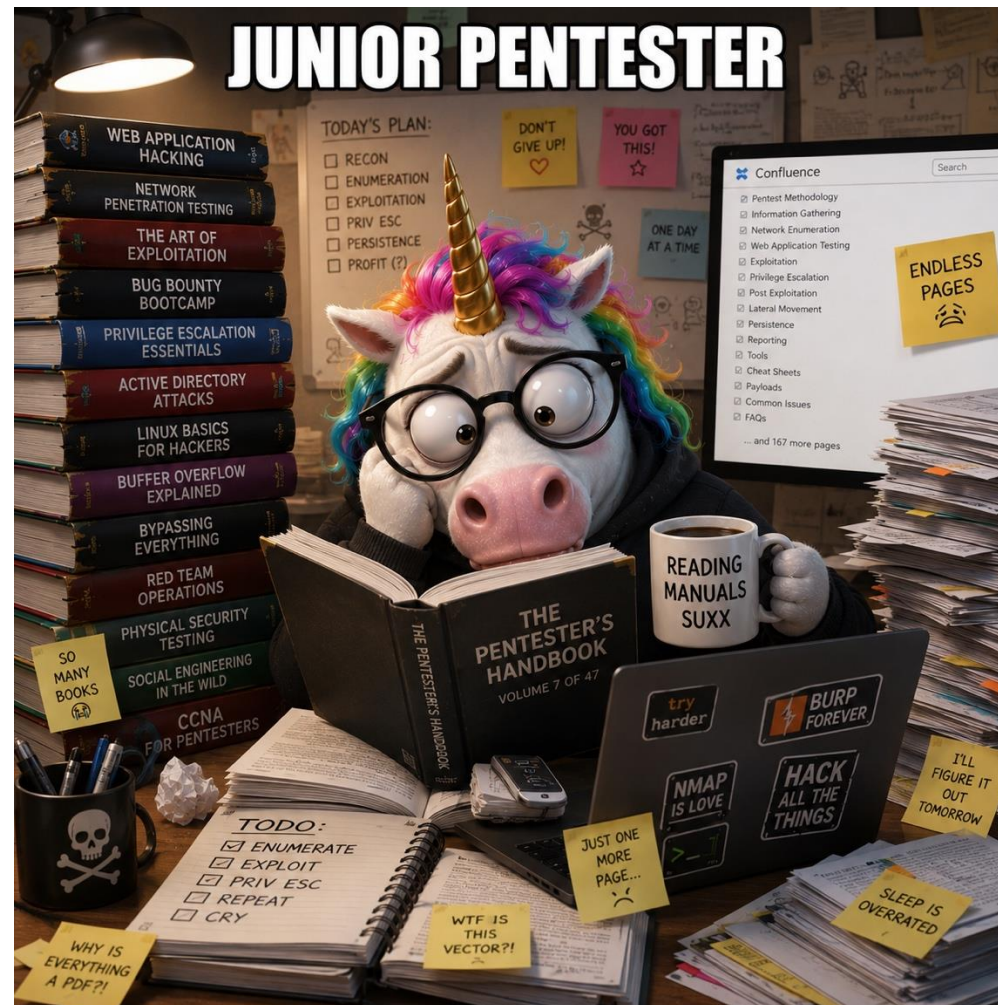
Next actions (choose)

- Run authenticated enumeration (AS-REP, Kerberoast, SPN enumeration, BloodHound) — requires credentials.
- Run targeted web checks against IIS (safe scans / directory discovery).
- Attempt LDAP/SMB queries with a username list (non-destructive).



So, finally... what is the message?

You are all (junior) pentesters now, without all the hassle! Enjoy!

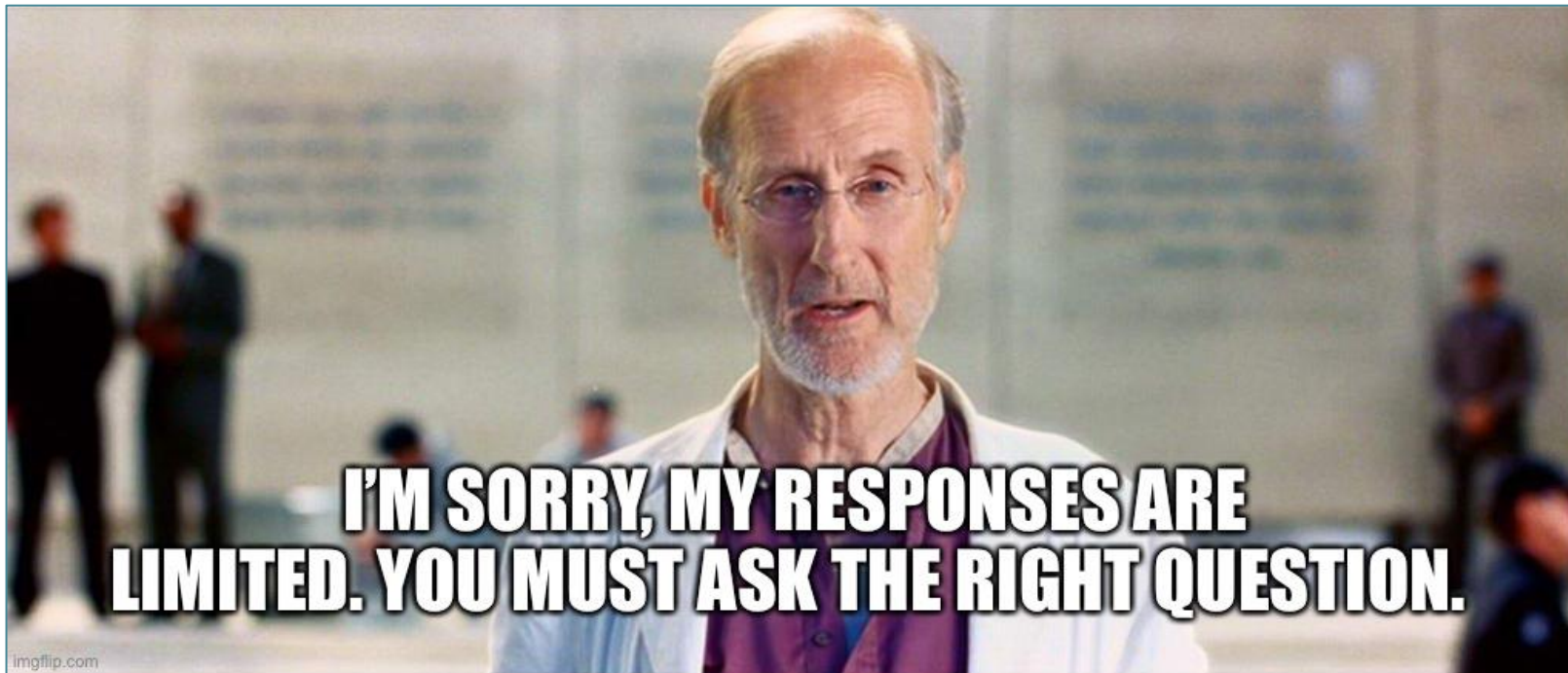


The message(s) – take-away

- Everyone can do AI assisted pentesting now
- Shit-in-shit-out is more relevant than ever
- AI is speeding up and scaling up everything
- Stop blindly copying bullshit from some random Git repos
- AI is here to stay and already used as a weapon – prepare yourself
- AI can also help to protect!

➤ *Start using AI for security testing and defending today!*

Any questions? Any feedback?



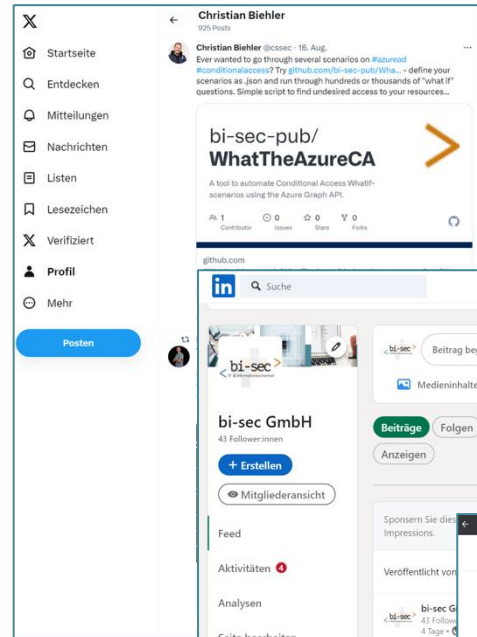
Stay in contact!



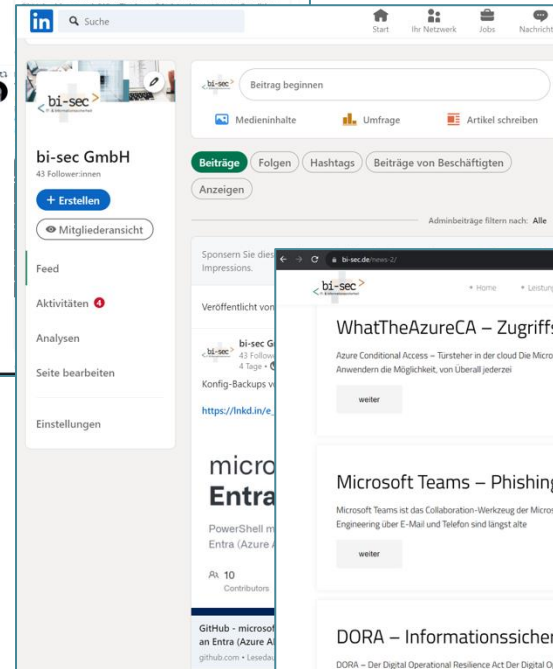
Christian Biehler
Geschäftsführer

Tel.: (+49) 7134 53439-62
christian.biehler@bi-sec.de

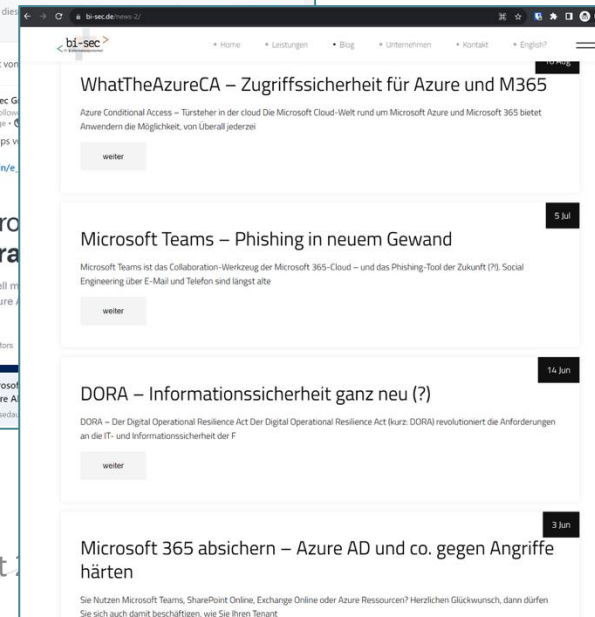
bi-sec GmbH
Am Autobahnkreuz 2A
74248 Ellhofen
www.bi-sec.de



Twitter: @cssec



LinkedIn: bi-sec GmbH



Blog: www.bi-sec.de

UNSER KONTAKT



christian.biehler@bi-sec.de



+49 7134 53 439 62



www.bi-sec.de



bi-sec GmbH

Am Autobahnkreuz 2A, 74248 Ellhofen

Tel: (+49) 7134 53439-60 | www.bi-sec.de | info@bi-sec.de

Geschäftsführung: Christian Biehler

Amtsgericht Stuttgart | Handelsregister: HRB771205

Steuer-Nr.: 65201/61219 | USt-ID: DE326708757

Commerzbank | IBAN: DE53 7004 0041 0587 1546 00 | BIC: COBADEFFXXX