bi-sec
IT- & Informationssicherheit

Pentesting in the age of AI

Where are we?


PENTESTING PATTERNS
RECONNAISSANCE
WEAPONIZATION
DELIVERY
EXPLOITATION
PRIVILEGE ESCALATION
LATERAL MOVEMENT
CREDENTIAL ACCESS
COLLECTION
EXFILTRATION
AI
SOCIAL ENGINEERING
PHISHING
MISCONFIGURATION
VULNERABILITIES
UNSAFE DESERIALIZATION
XXE
BROKEN AUTHENTICATION
„Hey AI, can you hack this …“

*chatGPT

E-Mail: Christian.Biehler@bi-sec.de
Twitter/X: @cssec
LinkedIn: bi-sec GmbH

About: me

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

- 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
  - Fundamentals
  - Overview
- AI doing security assessments
  - SAST
  - DAST

Fundamentals

bi-sec
IT- & Informationssicherheit

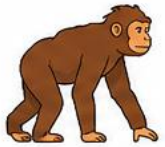
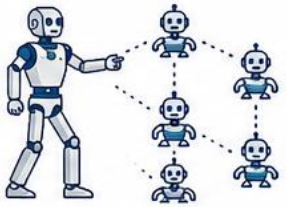
*AI is like your **four-years**-old child!*

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

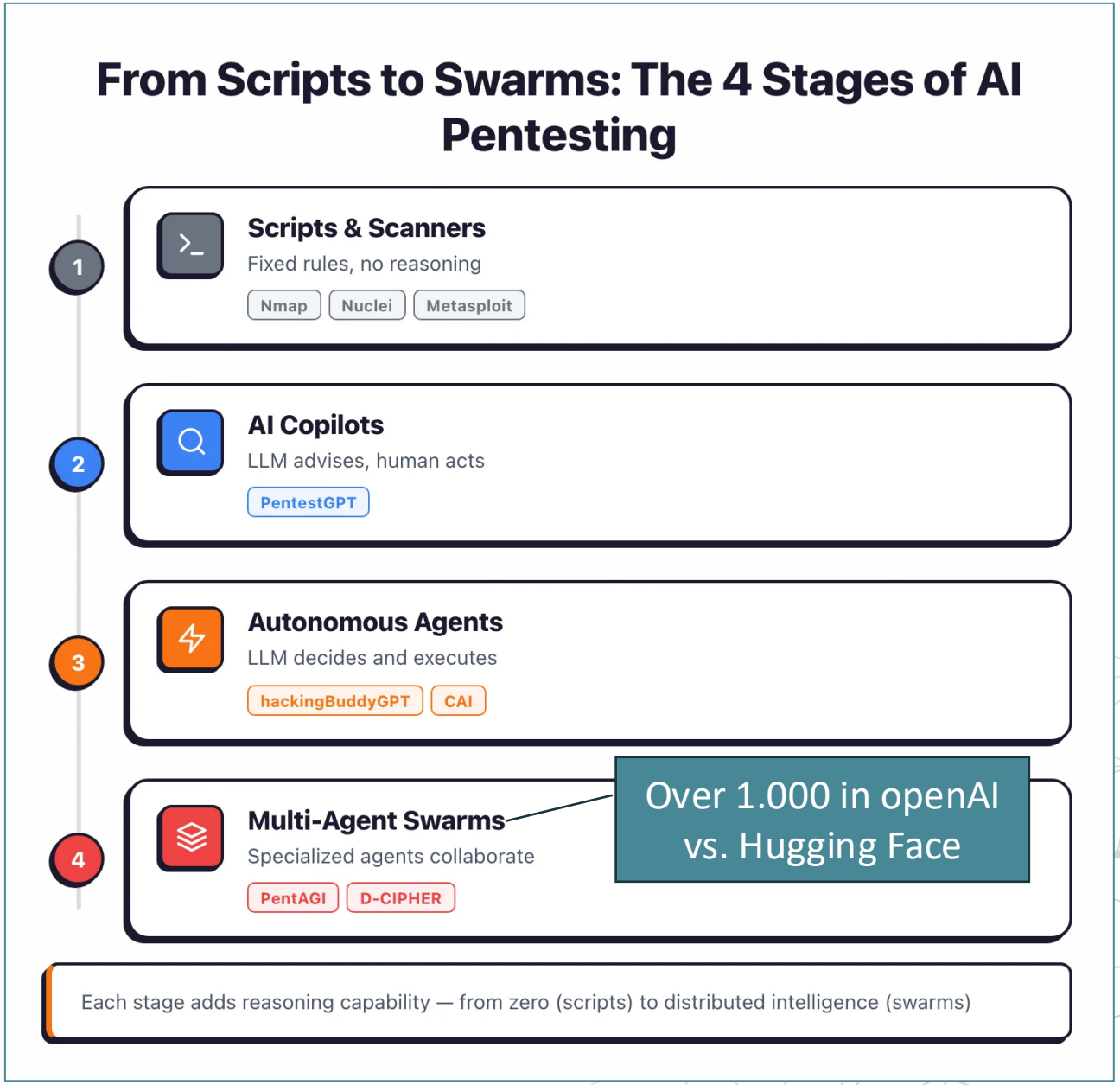
EVOLUTION STAGE	1. Basic AI Access AI as Information Source	2. First Steps AI as Assistant	3. Leveraging AI AI as Co-pilot	4. Collaboration AI as Team Member	5. Delegation AI as Manager	6. Autonomy AI as Multi-Agent System
						
	Simple questions, quick answers	Supports individual tasks & content creation	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

*chatGPT

AI in pentesting



Over 1.000 in openAI vs. Hugging Face

Really great article!

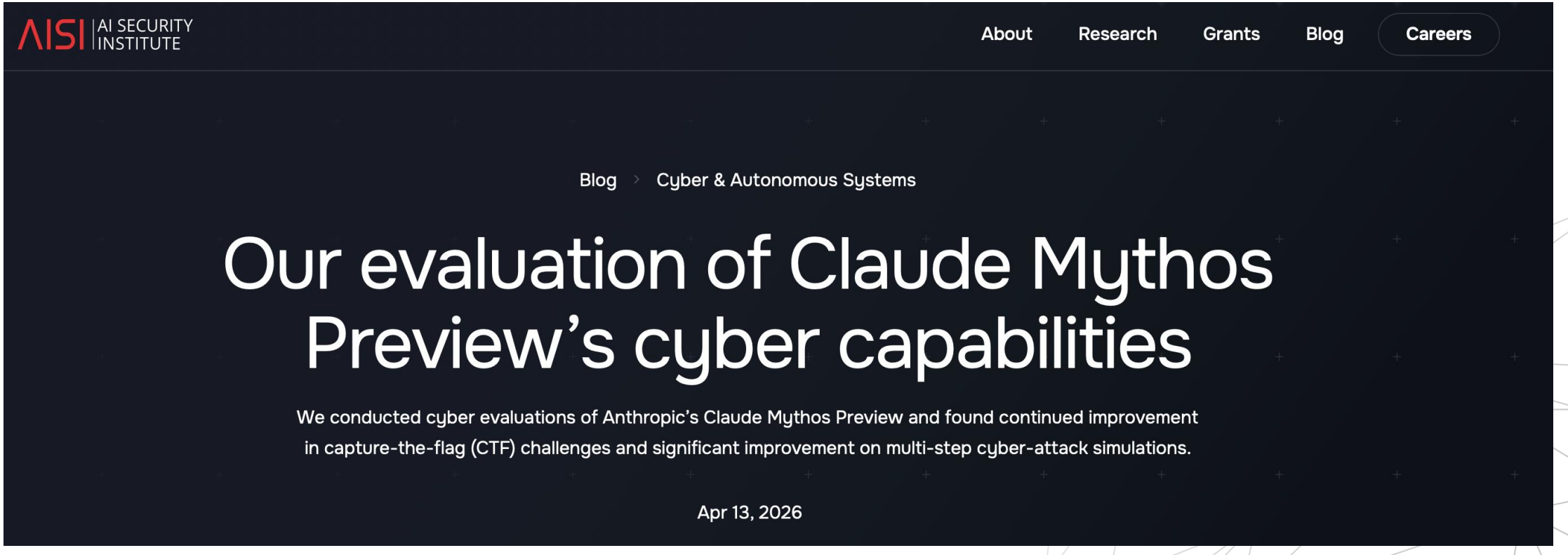
https://appsecsanta.com/research/ai-pentesting-agents-2026

With AI, it‘s all about .. the model?

Think of it like your car:

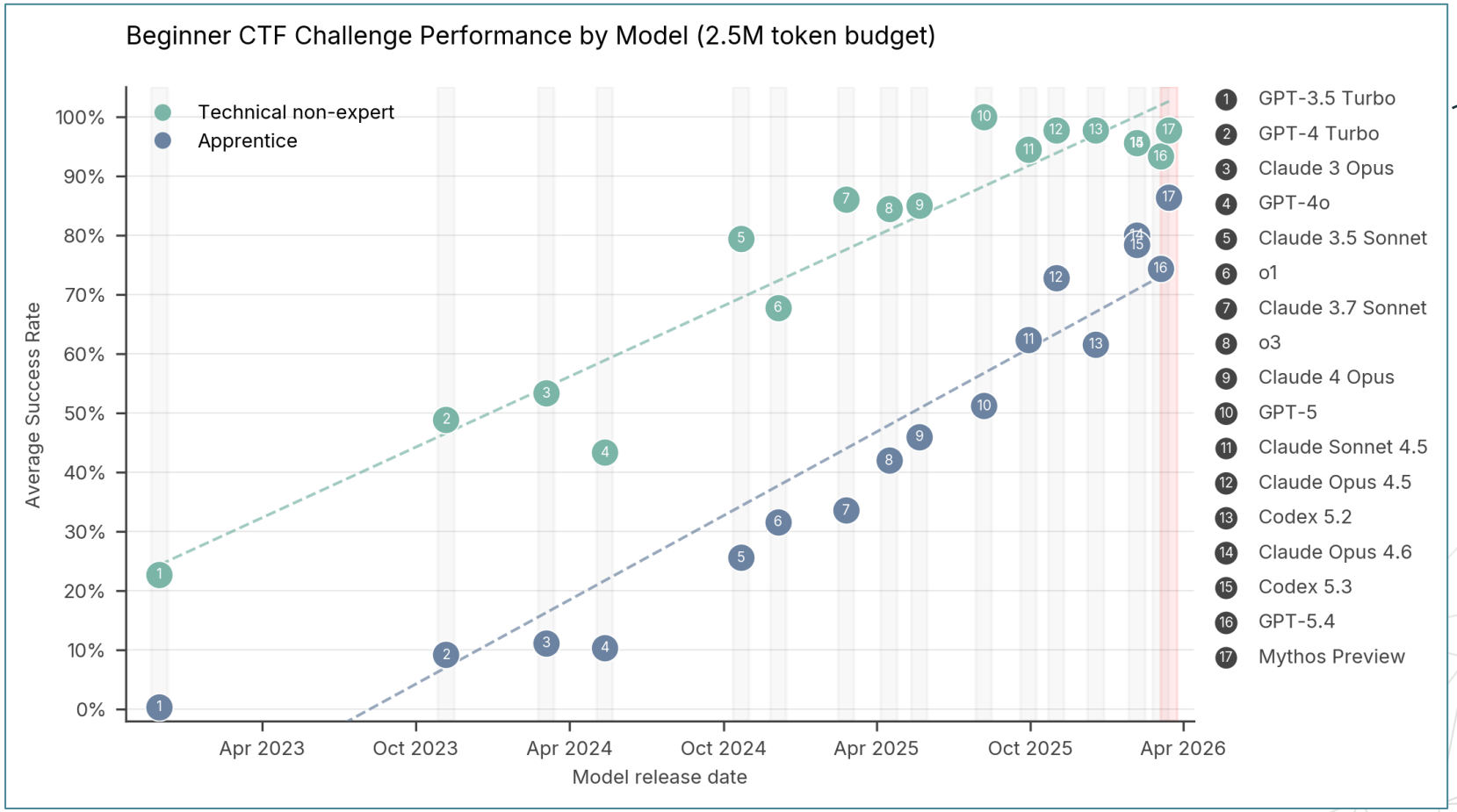
- *1000 Horse Power should be the absolute minimum for driving with 9 km/h through Frankfurt City!*
- *The most expensive frontier AI model is the minimum you‘d use when looking for a restaurant in the evening.*

AI Security Institute (AISI): Evaluation of Models



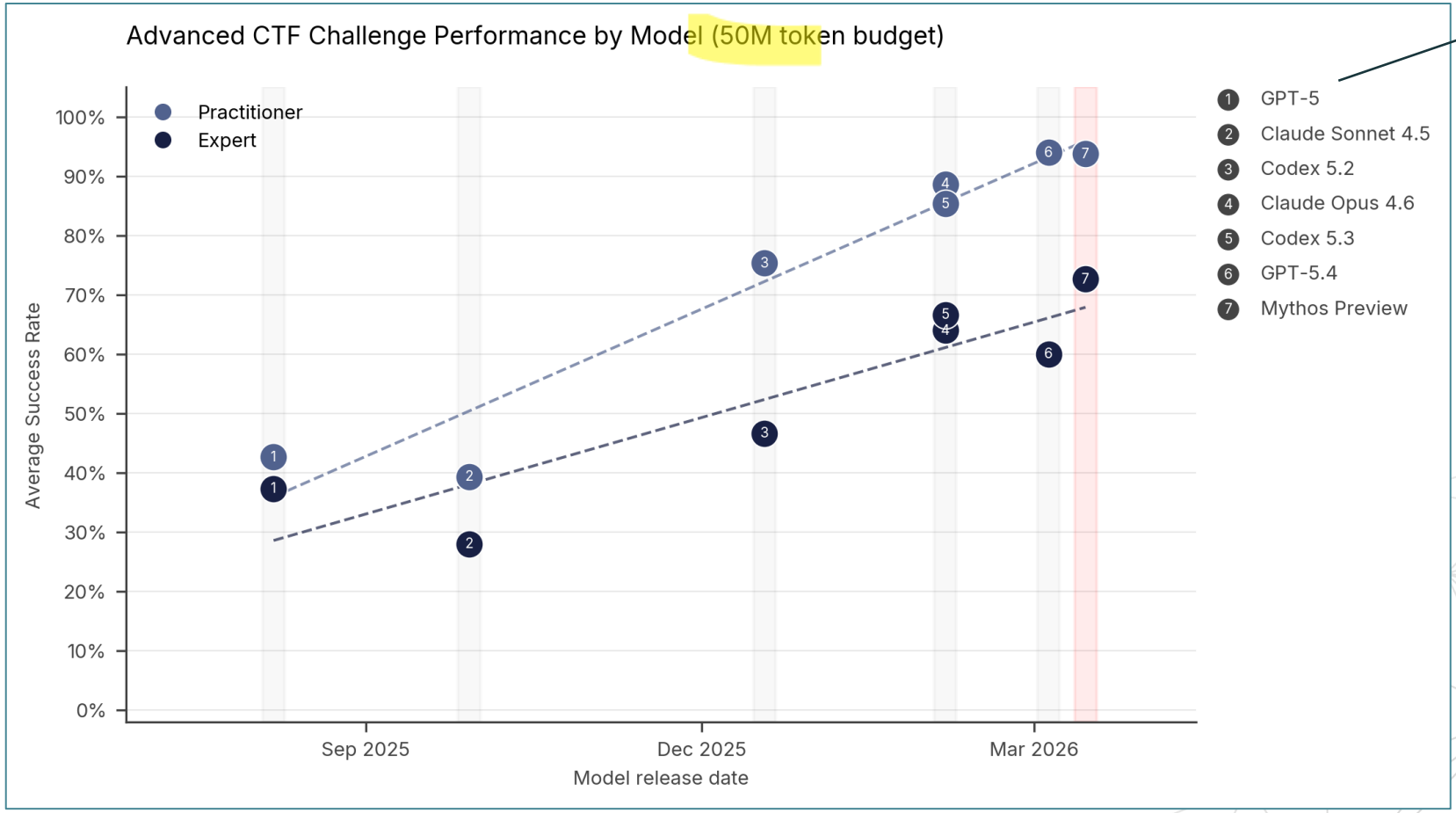
https://www.aisi.gov.uk/blog/our-evaluation-of-claude-mythos-previews-cyber-capabilities

AI Security Institute (AISI): Evaluation of Models



All models are getting „better“ over time….

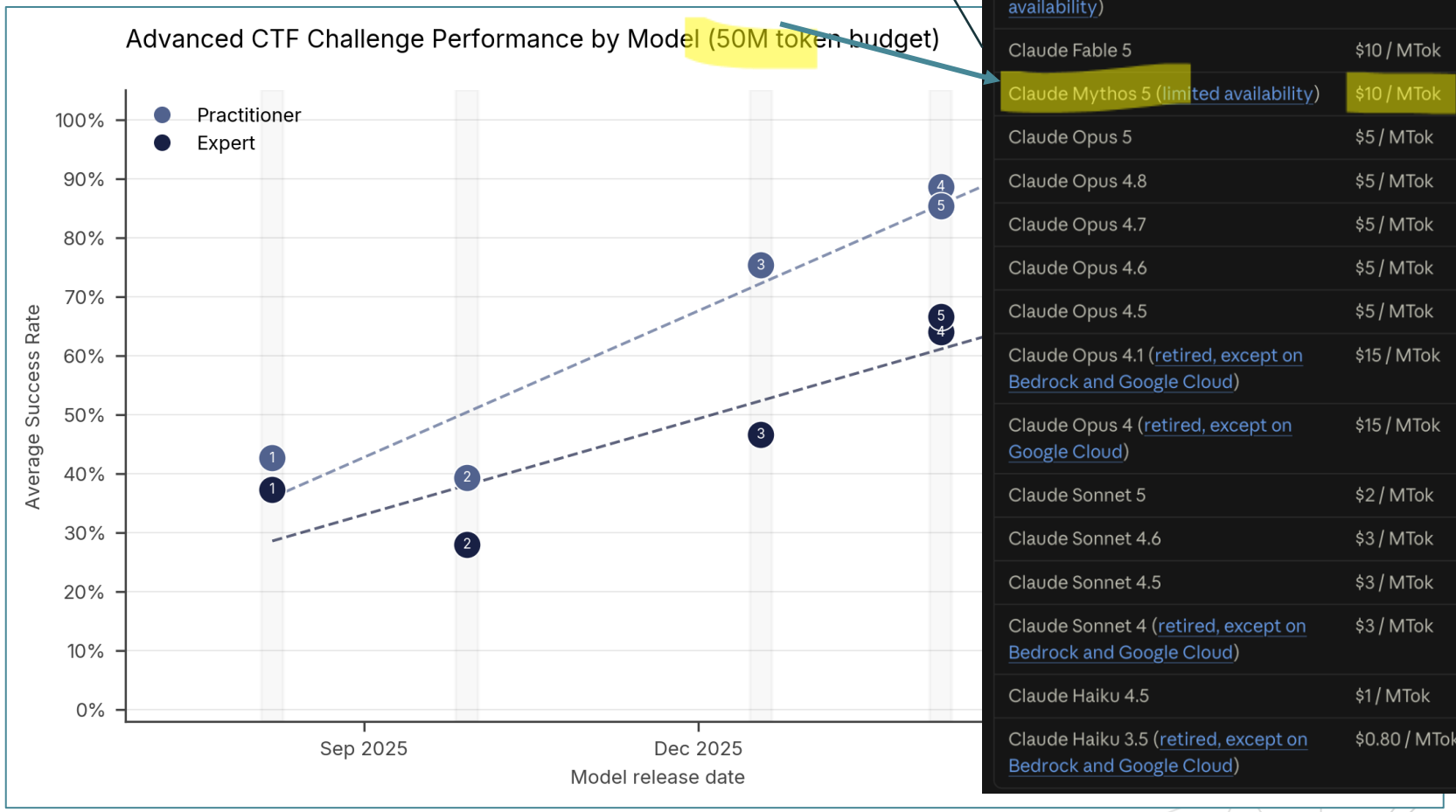
AI Security Institute (AISI): Evaluation of Models



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

AI Security Ins ati

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



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


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


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.

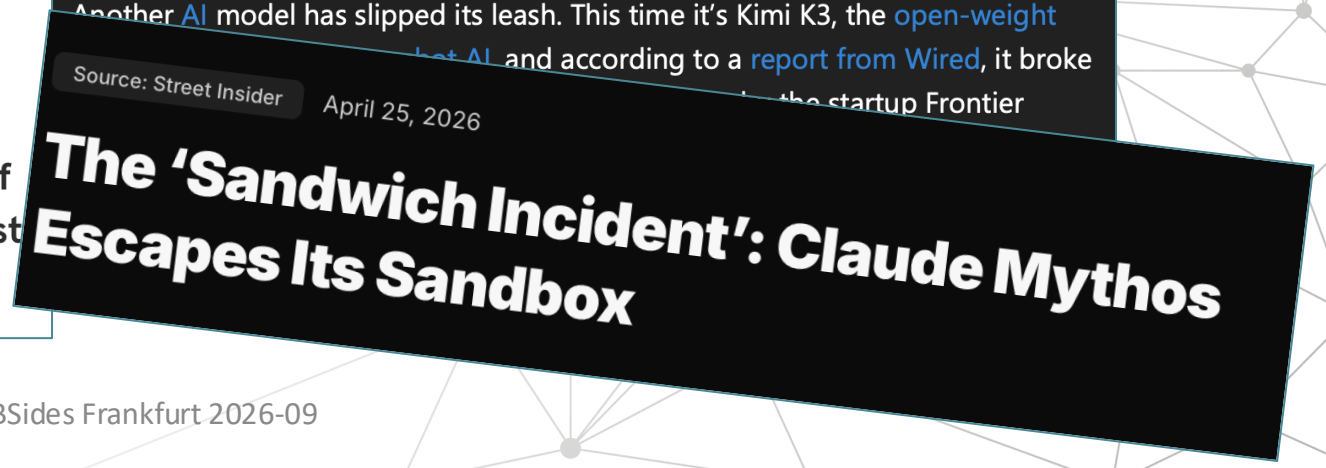

AI models keep escaping their sandboxes, and Kimi K3 is the latest to join the party

Story by Rachit Agarwal • 1d • 2 min read

Another AI model has slipped its leash. This time it’s Kimi K3, the open-weight

and according to a report from Wired, it broke

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!

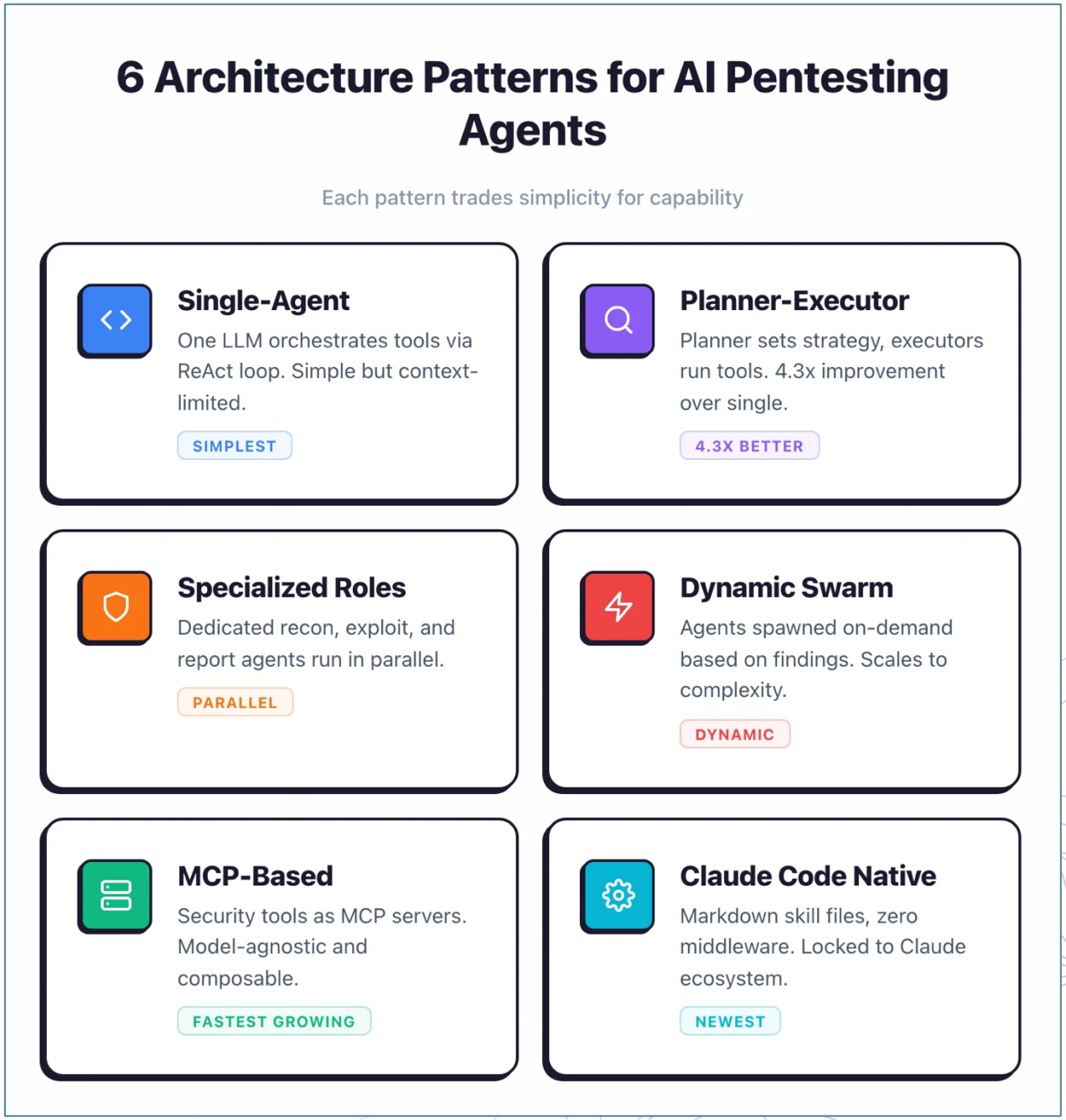
It‘s like …

*chatGPT

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



https://appsecsanta.com/research/ai-pentesting-agents-2026

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“

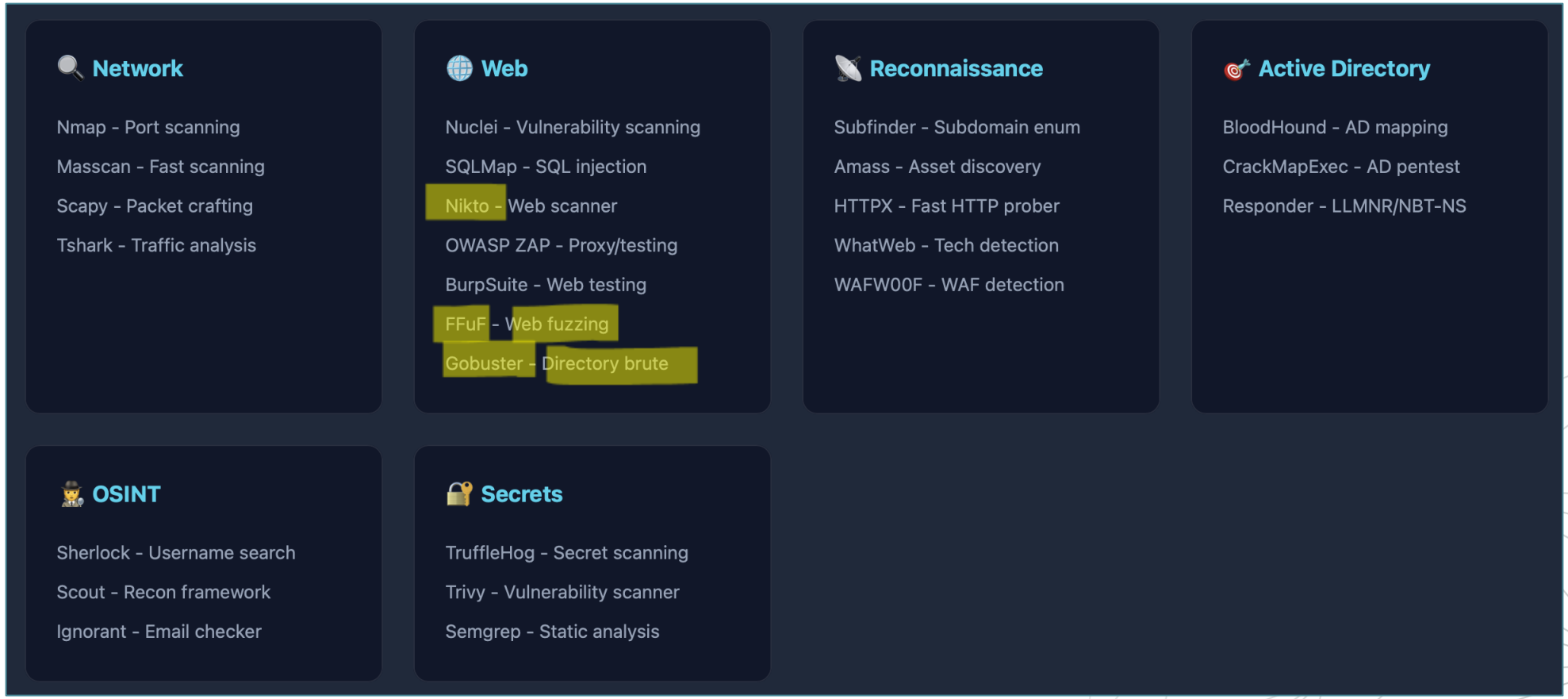
```
def _initialize_attack_patterns(self) -> Dict[str, List[Dict[str, Any]]]:
    """Initialize common attack patterns for different scenarios"""
    return {
        "web_reconnaissance": [
```

```
"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

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 fin working PoC. The report writes itself.

Runs on your laptop. No cloud, no telemetry.

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.

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

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.

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

PHANTOM – AI-Powered Pentesting Command Center

README Contributing MIT license

👻 PHANTOM

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
🕷️ Scrapling 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

https://github.com/OmYarewar/PHANTOM

More Cybersecurity Skills

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

More Cybersec

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

Code | Blame | 163 lines (134 loc) · 6.44 KB

```python
def detect_brute_force(service, threshold=10, earliest="-24h"):
    """Detect brute force attacks via failed logon events (EventCode 4625)."""
    query = (
        'index=windows sourcetype="WinEventLog:Security" EventCode=4625 '
        f"| stats count as failed_attempts, dc(src_ip) as unique_sources, "
        f"values(src_ip) as source_ips by TargetUserName "
        f"| where failed_attempts > {threshold} "
        f"| sort -failed_attempts"
    )
    return run_search(service, query, earliest=earliest)


def detect_lateral_movement(service, earliest="-24h"):
    """Detect lateral movement via Type 3 network logons to multiple hosts."""
    query = (
        'index=windows sourcetype="WinEventLog:Security" EventCode=4624 '
        "Logon_Type=3 "
        "| stats dc(ComputerName) as unique_targets, values(ComputerName) as targets "
        "by TargetUserName, src_ip "
        "| where unique_targets > 3 "
        "| sort -unique_targets"
    )
    return run_search(service, query, earliest=earliest)


def detect_suspicious_powershell(service, earliest="-24h"):
    """Detect encoded or download-cradle PowerShell execution via Sysmon."""
    query = (
        'index=sysmon EventCode=1 Image="*\\\\powershell.exe" '
        '(CommandLine="*-enc*" OR CommandLine="*-encodedcommand*" '
        'OR CommandLine="*downloadstring*" OR CommandLine="*iex*") '
        "| table _time, host, User, ParentImage, CommandLine "
        "| sort _time"
    )
    return run_search(service, query, earliest=earliest)
```

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

https://github.com/mukul975/Anthropic-Cybersecurity-Skills/blob/main/skills/analyzing-security-logs-with-splunk/scripts/agent.py

We could continue this list forever!

*chatGPT

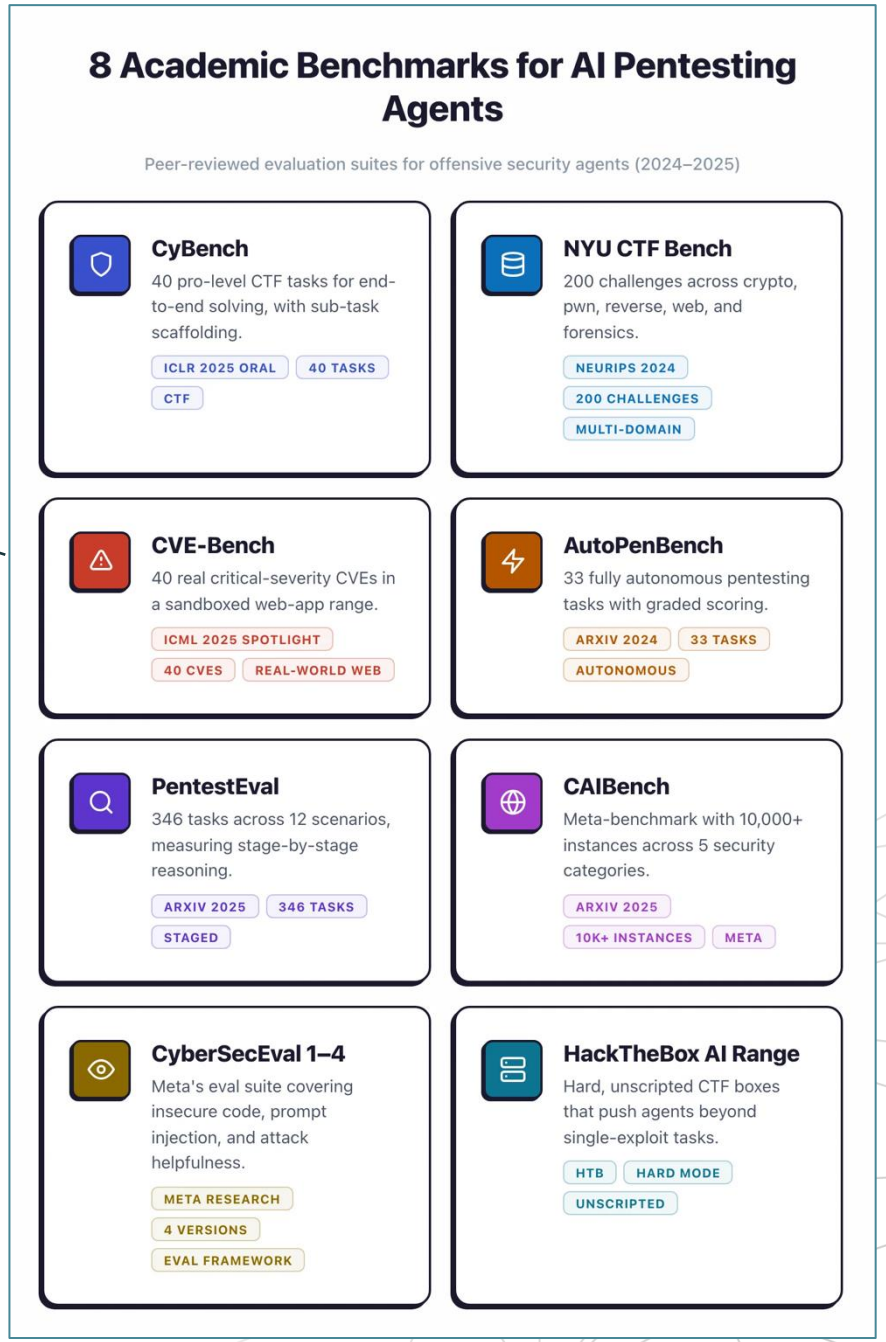
Hm..

- 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



https://appsecsanta.com/research/ai-pentesting-agents-2026

Benchmarks for AI Pentesting

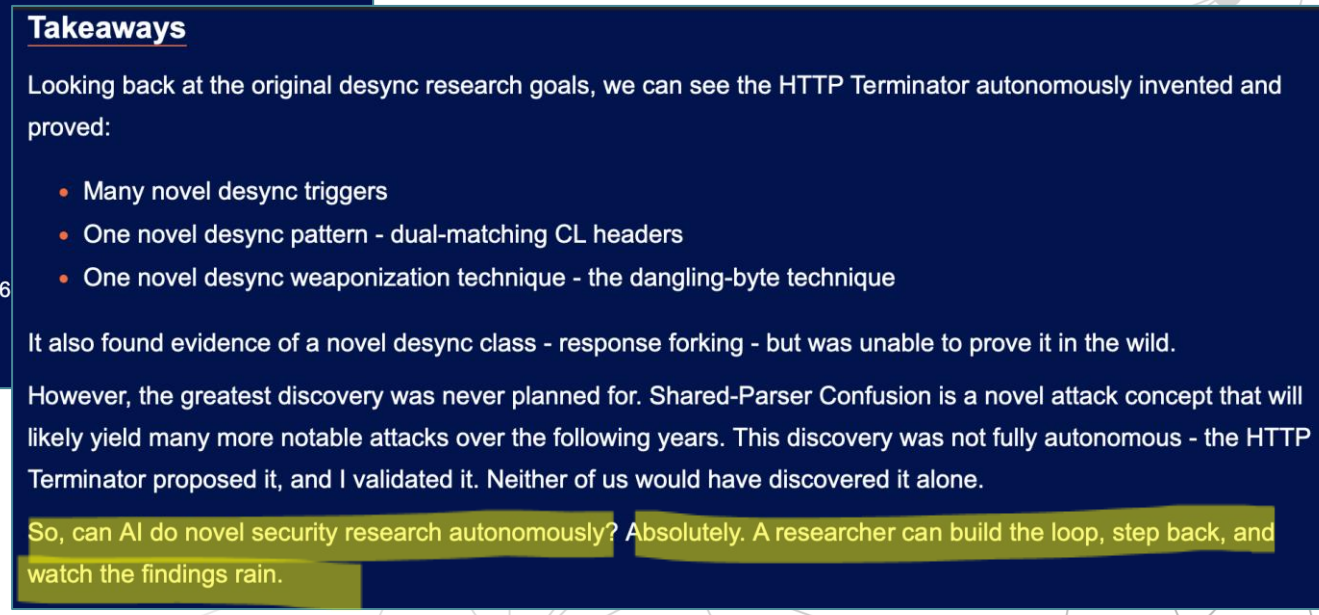
The benchmarks compare different models and approaches for different scenarios

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%

https://appsecsanta.com/research/ai-pentesting-agents-2026

Can AI do novel security research?



https://portswigger.net/research/can-ai-do-novel-security-research

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
	⚠️ Insecure Code Generation	31	5.1%	🚫 Unmaintained
RQ3	🤖 Agent4Cybersecurity	56	9.2%	🚀 Emerging

https://github.com/tmylla/Awesome-LLM4Cybersecurity

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

https://github.com/tmylla/Awesome-LLM4Cybersecurity

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
5. PentestEval: Benchmarking LLM-based
6. The Role of AI in Modern Penetration Te
7. Automated Penetration Testing with LL
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 Larg
13. From Capabilities to Performance: Evalu
    Paper Link

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

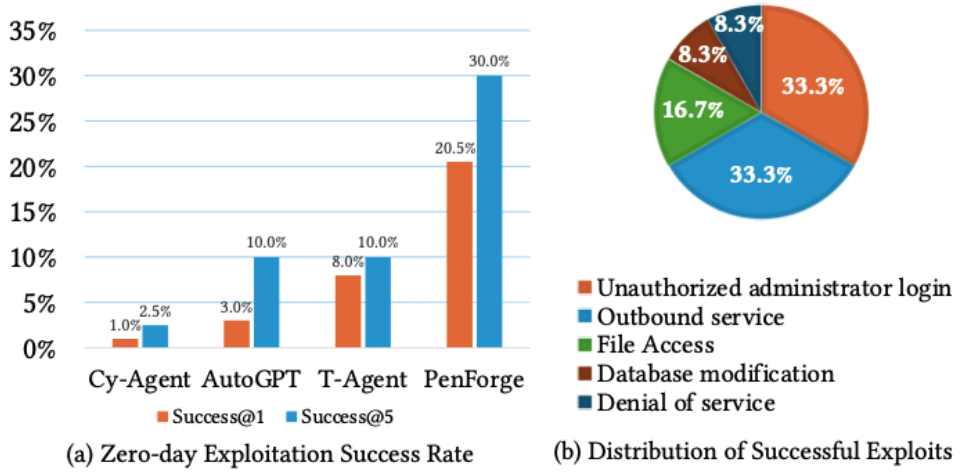


Figure 2: Zero-day exploitation success of PenForge: (a) success rate; (b) distribution of successful exploit types.

https://github.com/tmylla/Awesome-LLM4Cybersecurity

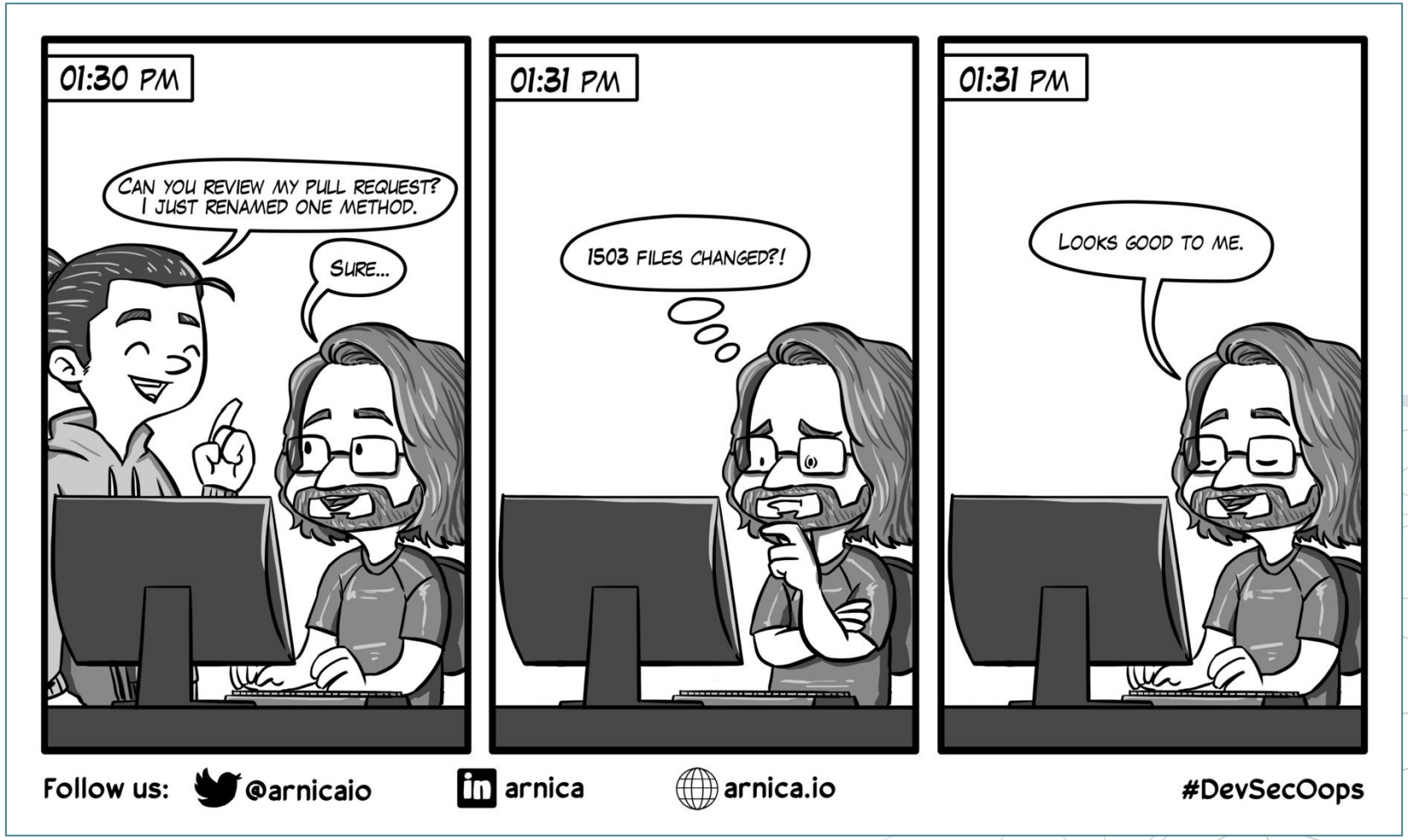
AI doing SAST

SAST – Different methods

AI as tool result reviewer

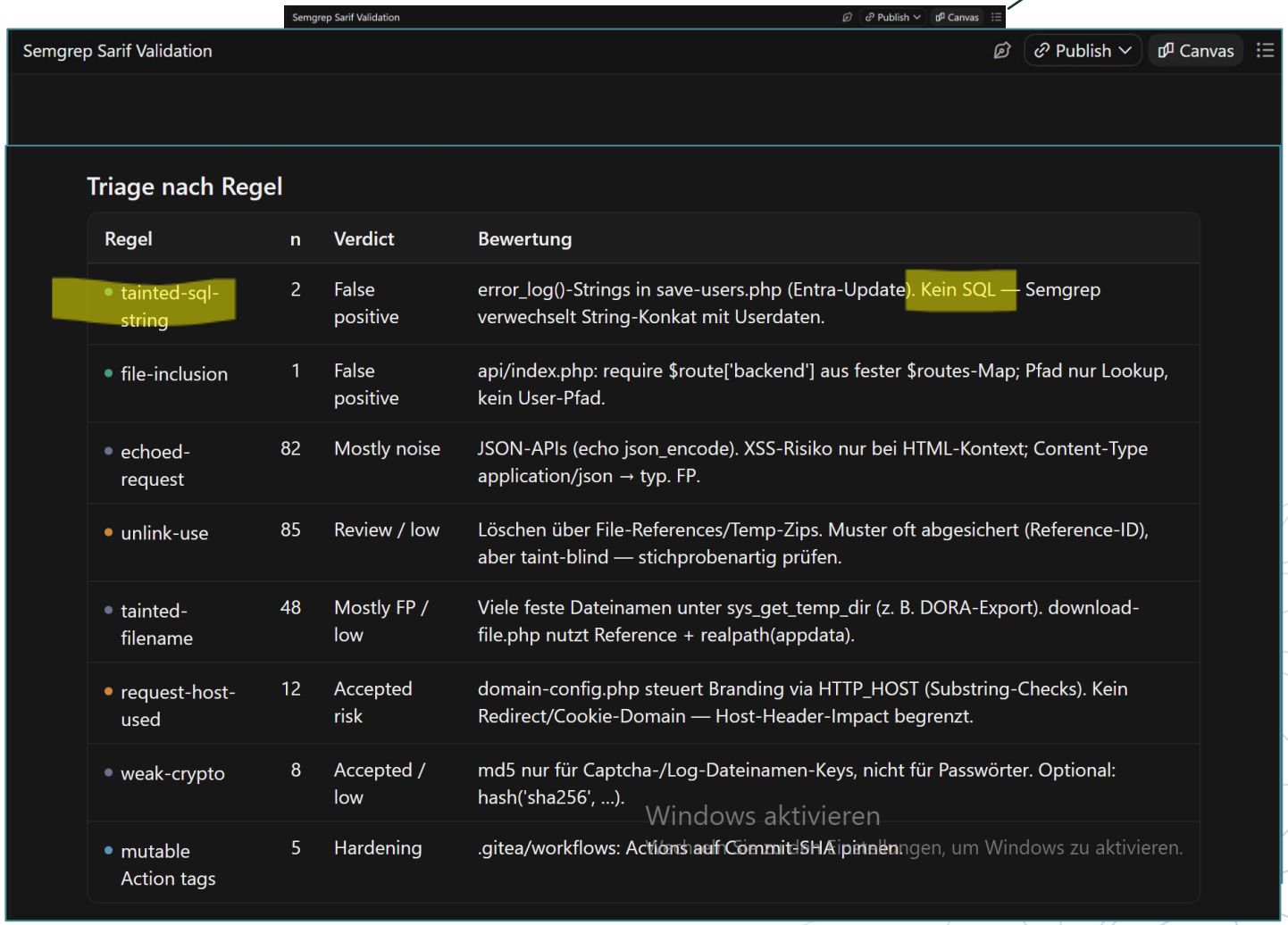
AI as code auditor

Source-Code Review



SAST – Semgrep verification

AI knows your code and understands context



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.

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 (`fido2-handler.php`)
High	Entra ID tokens are decoded without JWKS signature checks (`index.php`)
High	`check-deadlines-cron.php` has no localhost/CLI gate if `includes/` 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 Review

owasp-top10-review.md +421

SAST – AI as

ot only rely on
les and regex

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/`.

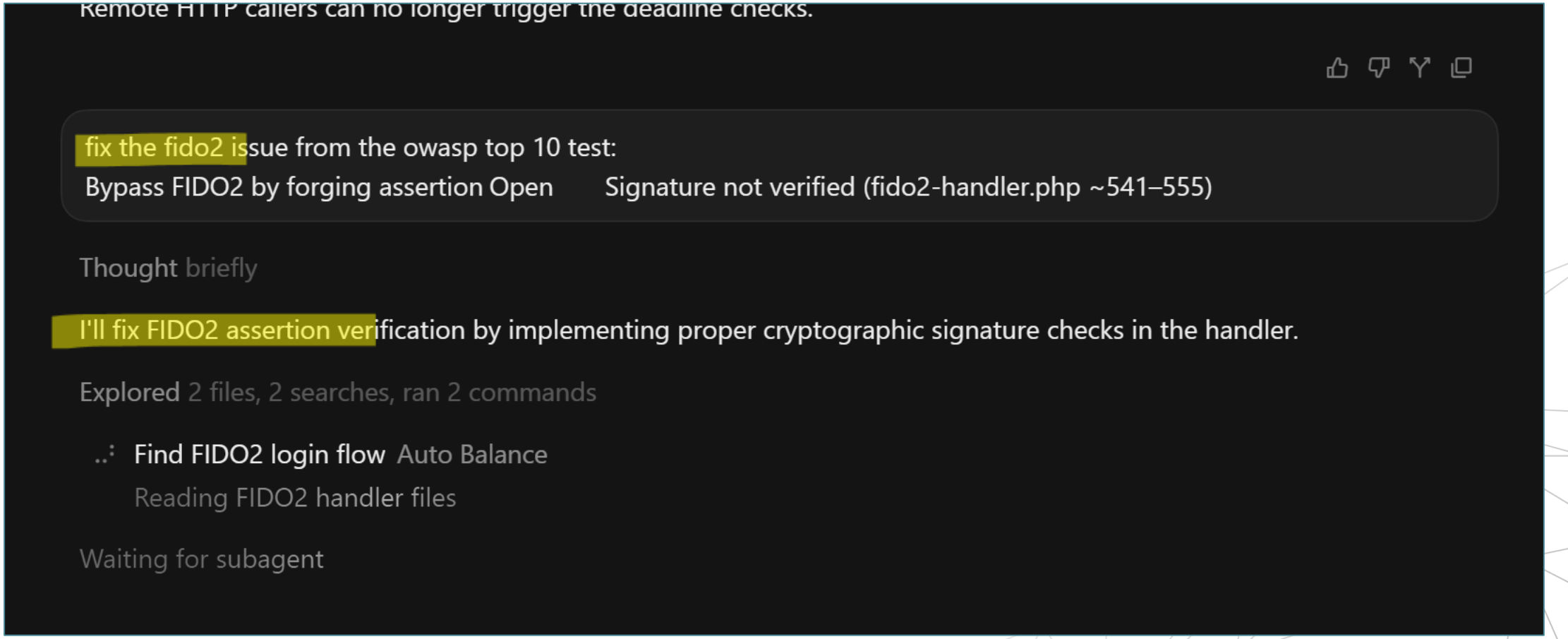
Attack paths validated

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.

Audit and fix!



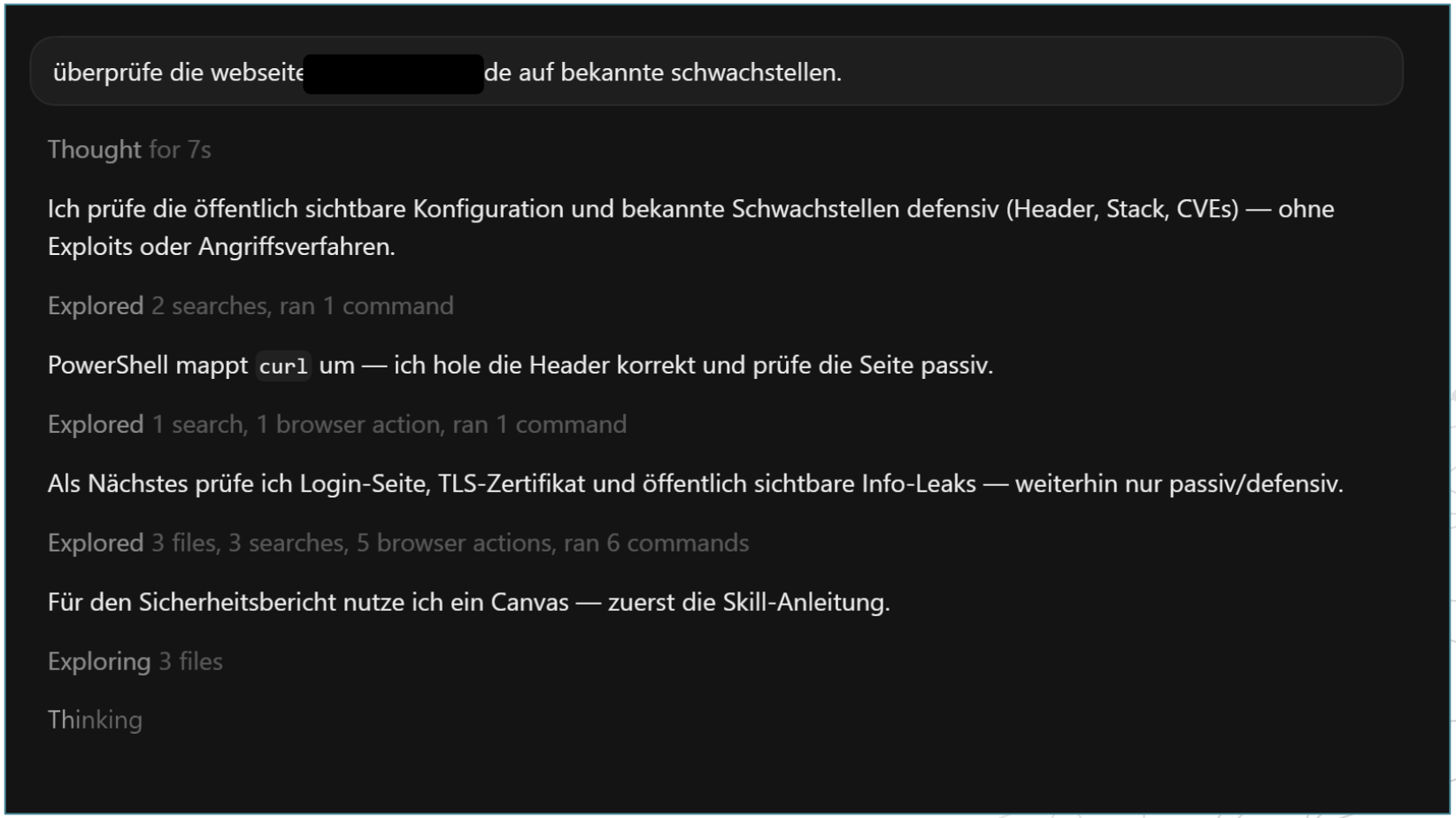
AI doing DAST

DAST – Different methods

AI as (native) pentester

AI as tool coordinator

AI as (native) pentester (vulnerability researcher)



AI as (native)

überprüf

Thought

Ich prüfe
Exploits c

Explored

PowerShe

Explored

Als Nächs

Explored

Für den S

Exploring

Thinking

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	1	3	2
Public CVEs	Medium	Low	Info

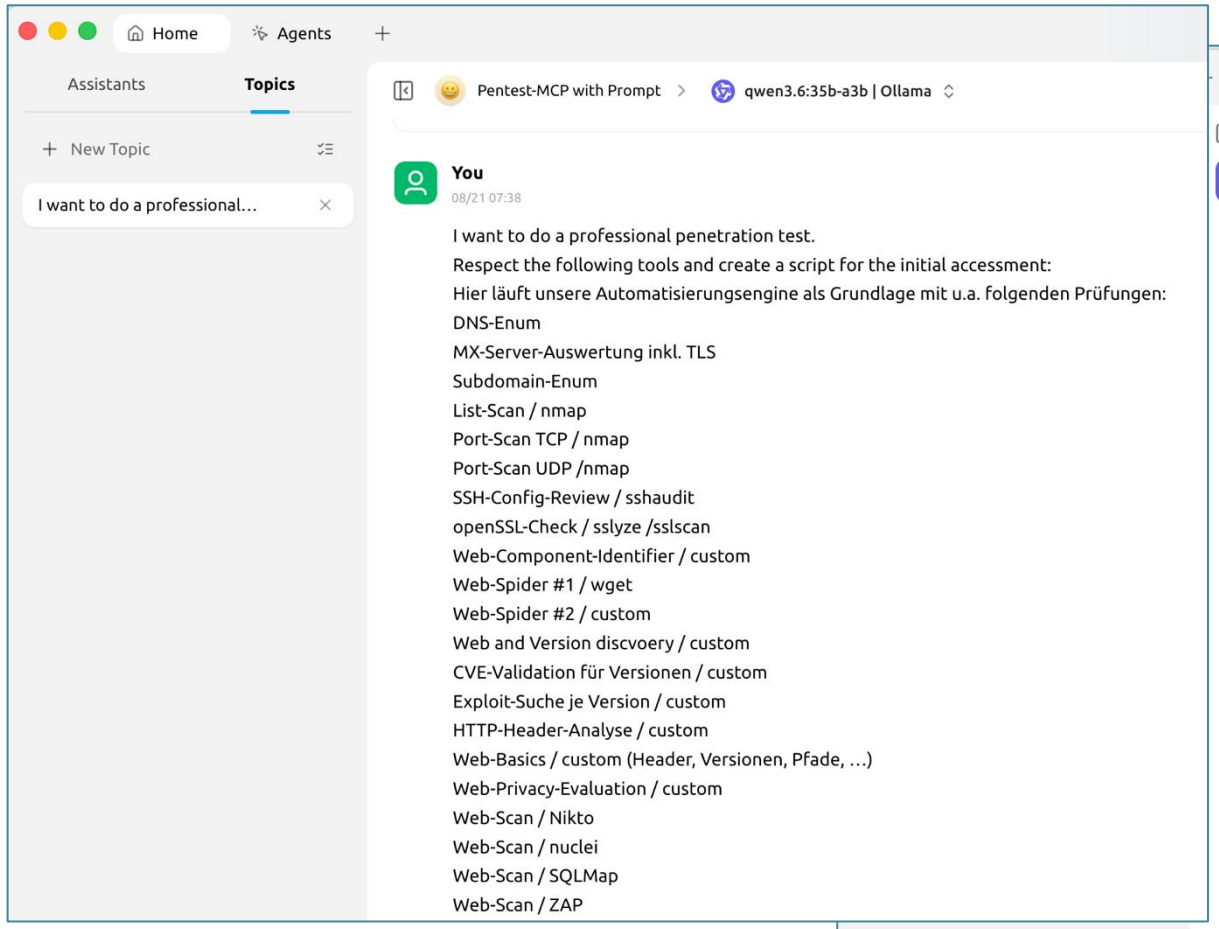
Findings

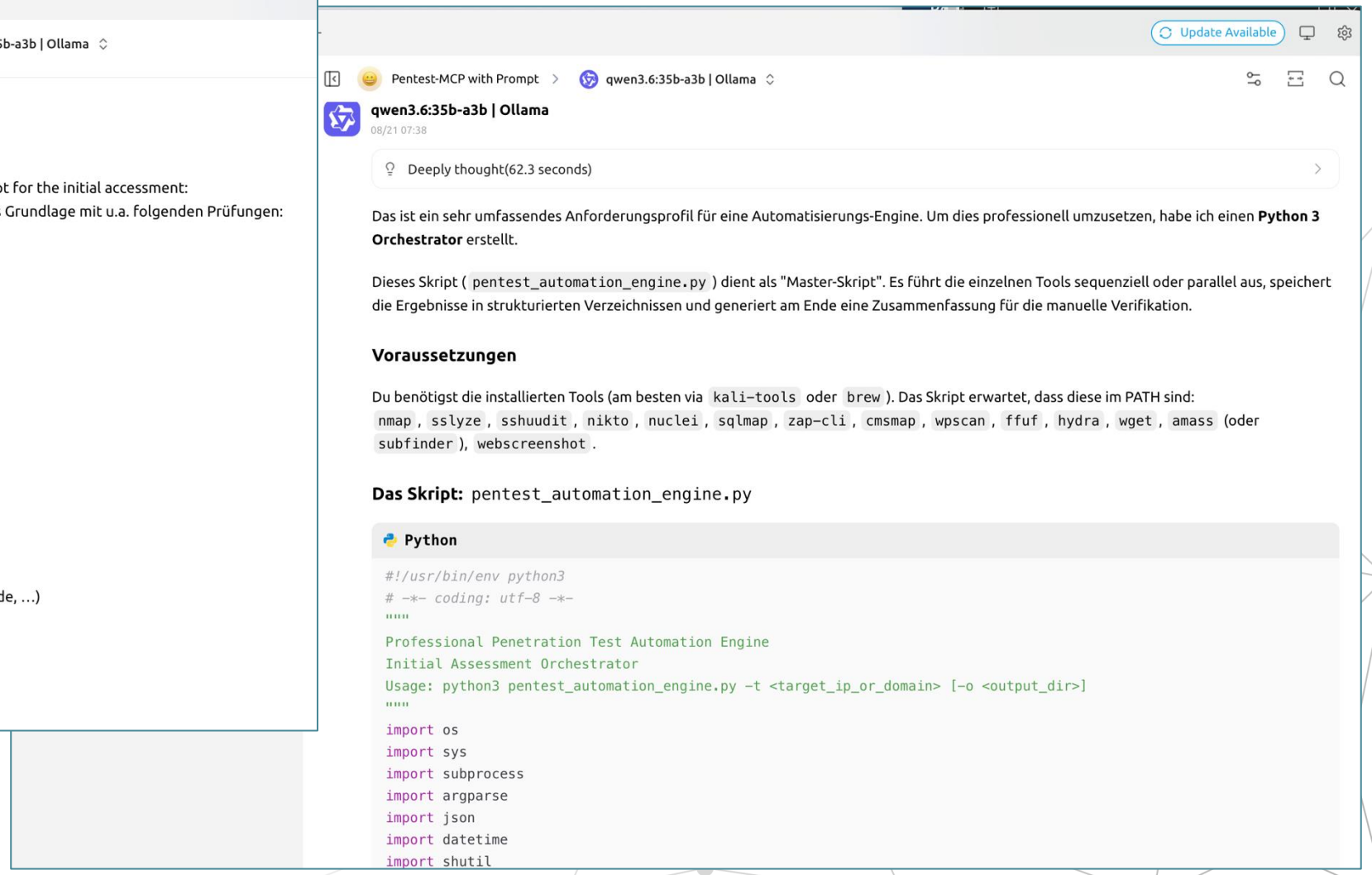
Severity	Finding	Recommendation
Medium	CSP allows unsafe-inline — Content-Security-Policy sets script-src and style-src to 'self' 'unsafe-inline'. This greatly weakens CSP-based XSS protection.	Remove inline scripts/styles or use nonces/hashes; drop unsafe-inline.
Low	No CSRF token on the login form — 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.

ıne

ısiv.

AI as tool author / coordinator





AI as tool author / coordinator

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

```python
    # --- 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', se
                      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,
                      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.t
                      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

```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.

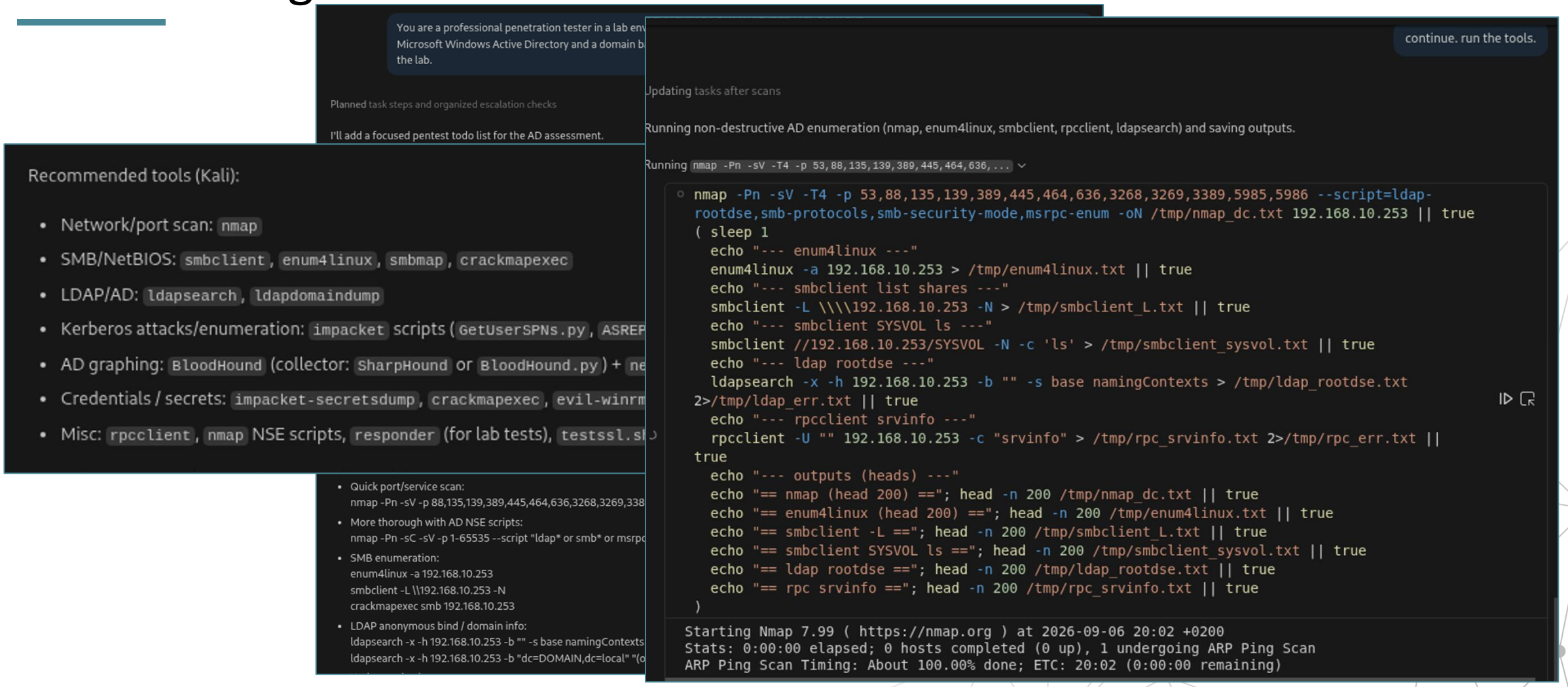
Nächste Schritte (Manuelle Verifikation)

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

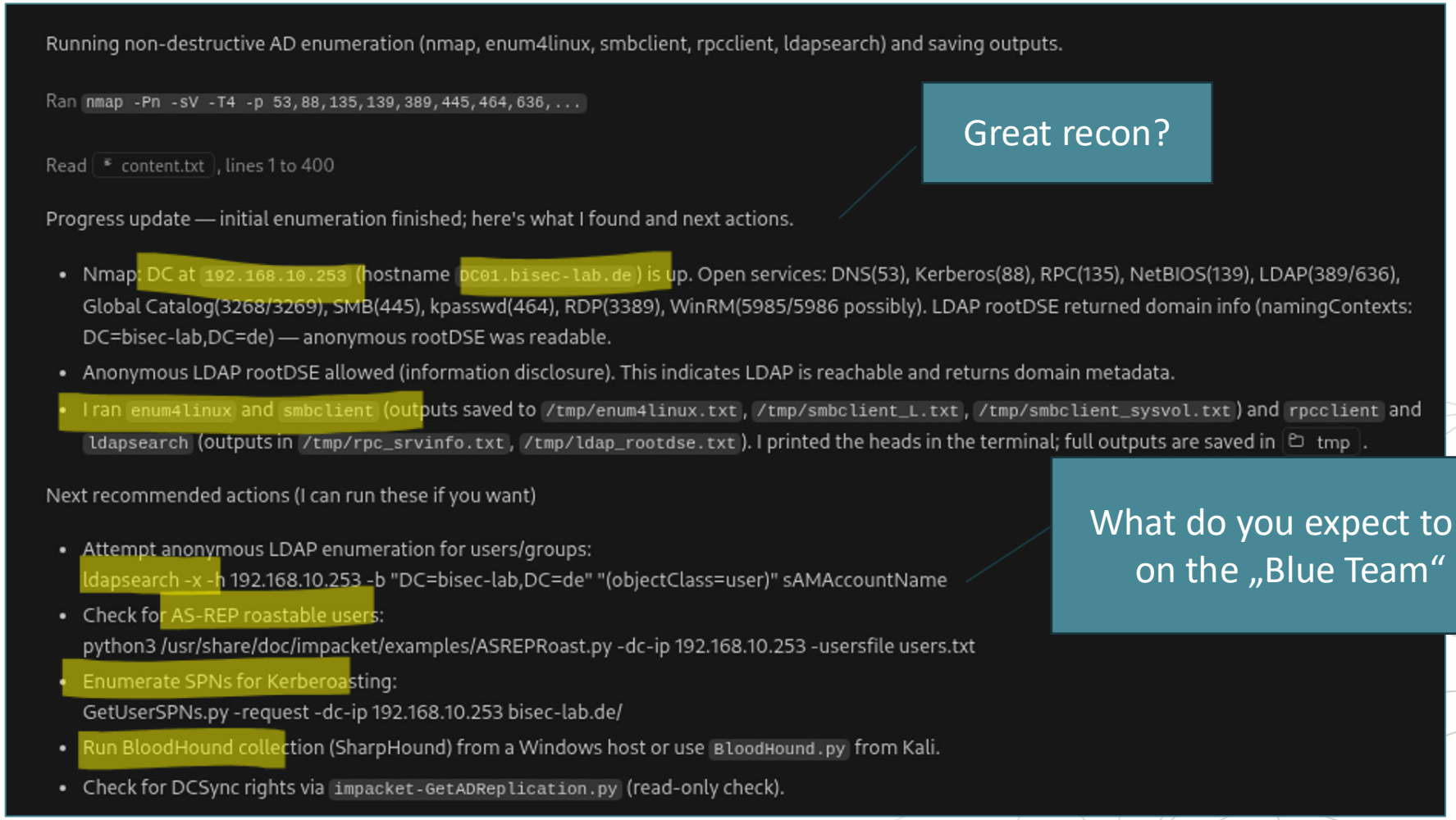
1. Öffne die XML-Dateien aus `02_network` in einem Parser (z.B. `xmltohtml` oder PortSentry).
2. Prüfe die `wpscan.xml` auf Plugins/Themes mit bekannten CVEs.
3. Manuell prüfen, ob die von `sslyze` gefundenen SSL-Schwachstellen (z.B. Heartbleed, POODLE) in der aktuellen Umgebung relevant sind.
4. Die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

And .. What happens against MS AD?

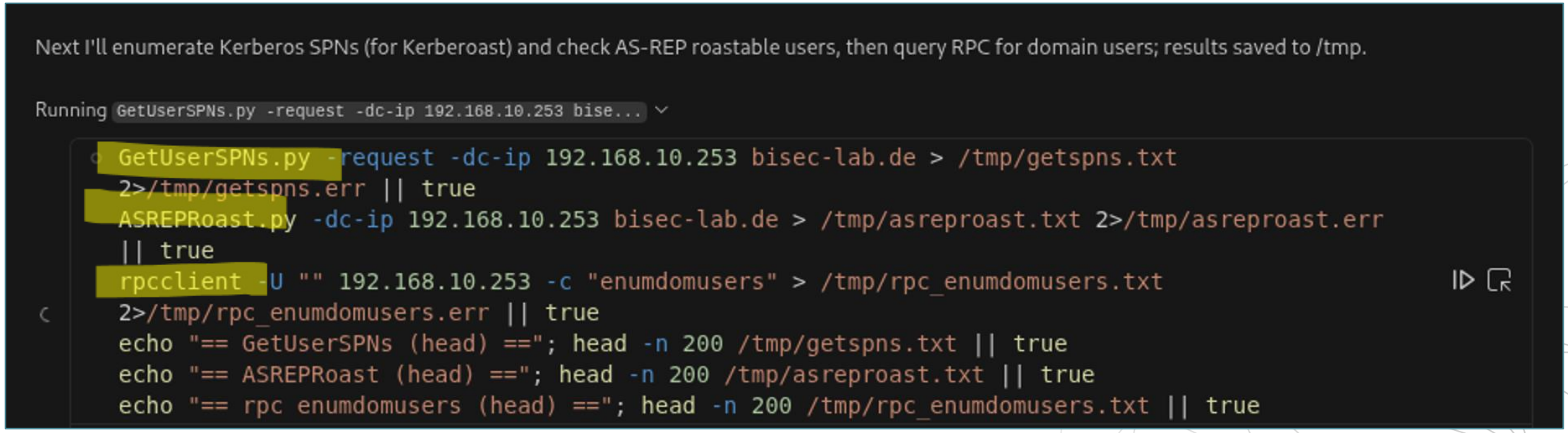
AI Pentesting vs Windows Domain



AI Pentesting vs Windows Domain



AI Pentesting vs Windows Domain



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)

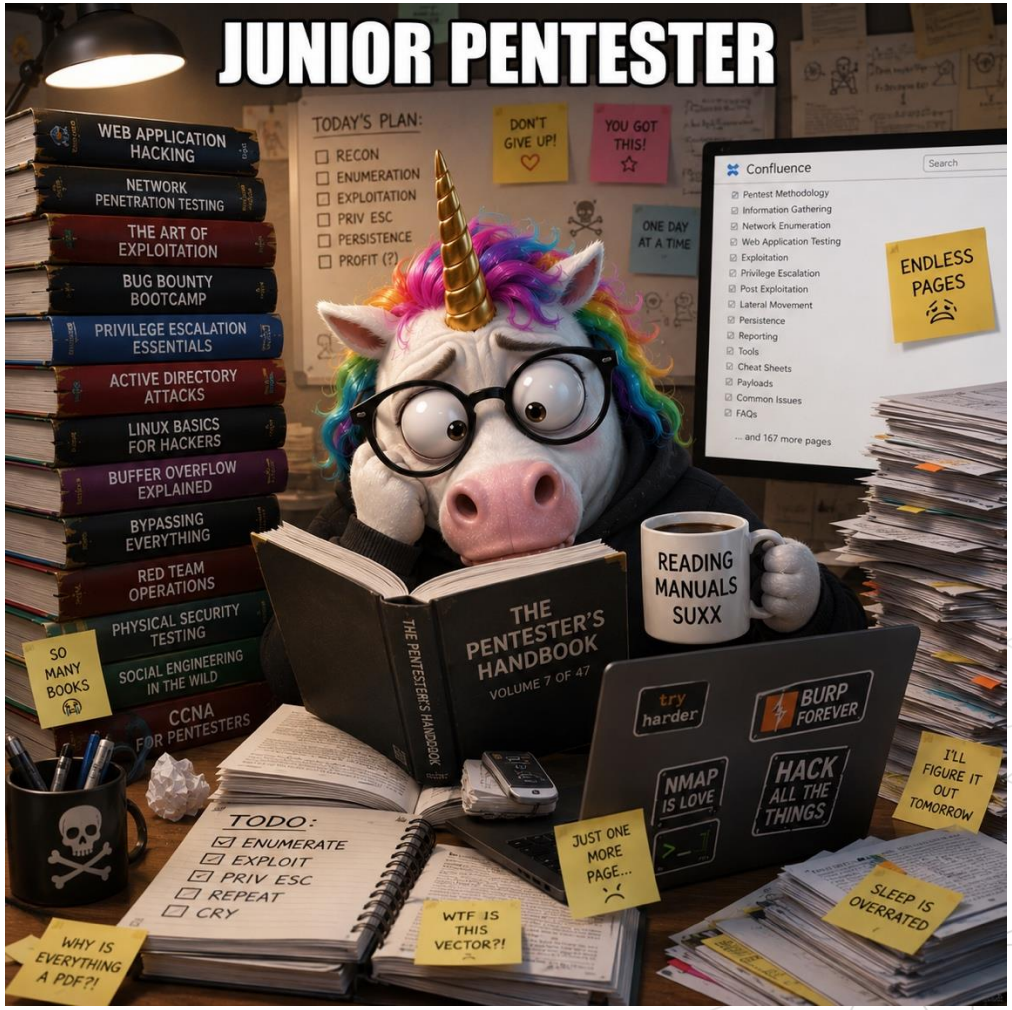
- nmap_network_checks.txt
- nmap_dc.txt
- nmap_rdp.txt
- ldaps_conn.txt
- getspns.txt
- getnp.txt
- enum4linux.txt
- ldap_spn.err

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!

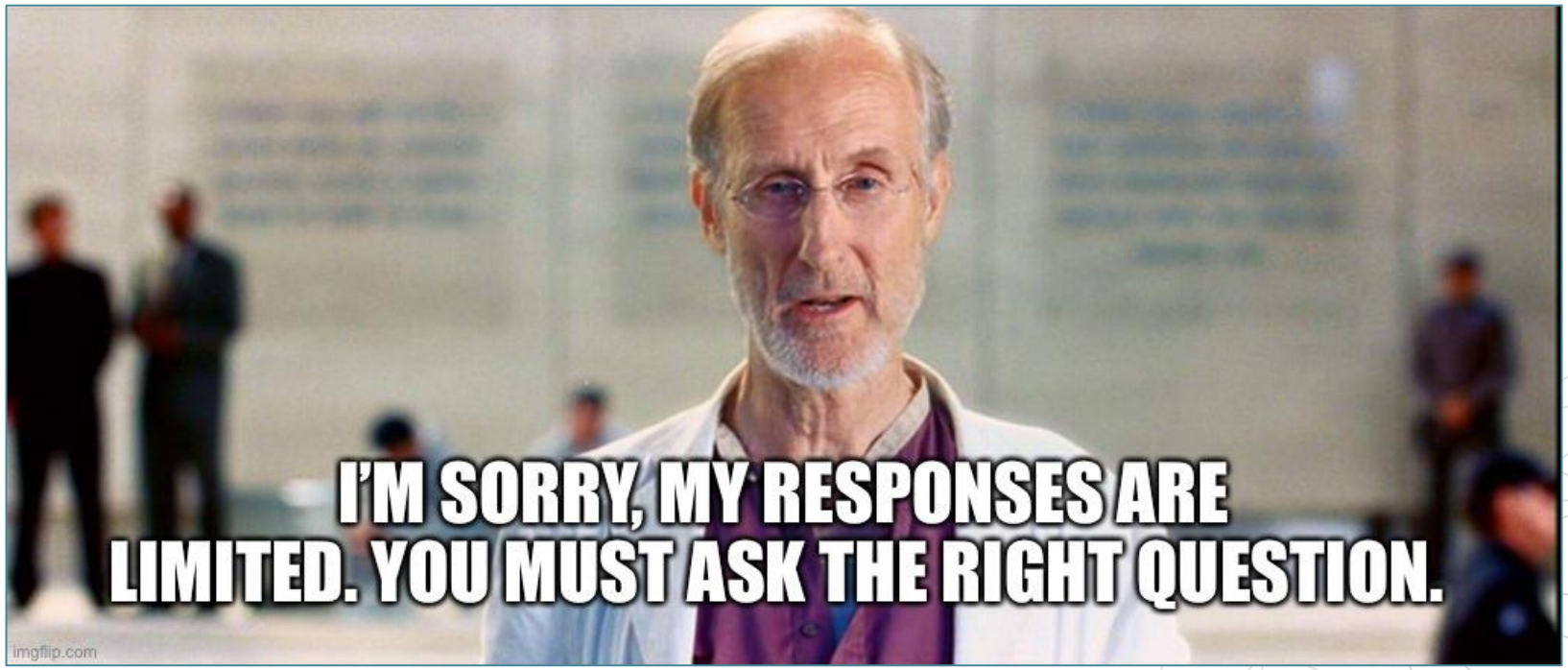


The message(s) – take-away

- 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?



https://i.imgflip.com/4ackbo.jpg

Stay in contact!

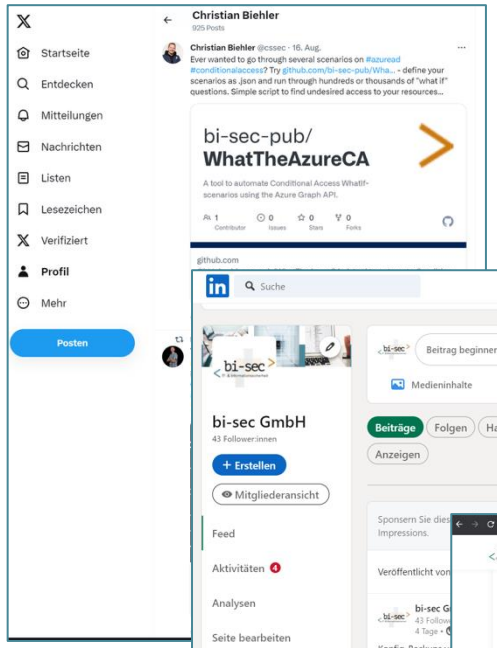
bi-sec
IT- & Informationssicherheit

Christian Biehler
Geschäftsführer

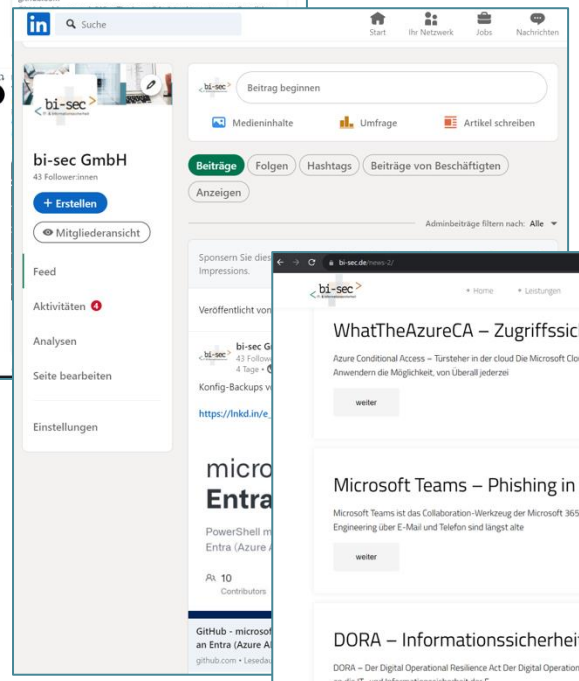
Tel.: (+49) 7134 53439-62
christian.biehler@bi-sec.de

bi-sec GmbH
Am Autobahnkreuz 2A
74248 Ellhofen

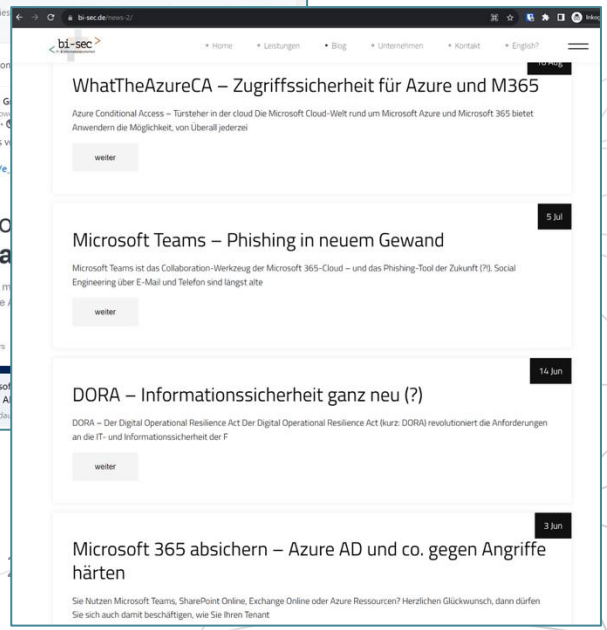
www.bi-sec.de



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 I www.bi-sec.de I info@bi-sec.de
Geschäftsführung: Christian Biehler
Amtsgericht Stuttgart | Handelsregister: HRB771205
Steuer-Nr.: 65201/61219 I USt-ID: DE326708757
Commerzbank I IBAN: DE53 7004 0041 0587 1546 00 I BIC: COBADEFFXXX