OH GREAT,

Another AI Talk

How Attackers Are Using LLMs to Recon Your Org

Black Hat 2026 · Sprocket Security

Sprocket Security

What We're Covering

01 — Reframing the Threat

02 — Automated OSINT at Scale

03 — Spear-Phishing at Scale

04 — Exploit Scaffolding

05 — The Defensive Framework

~35 min + Q&A

PART 1 OF 5

Reframing the Threat

What's changed, what hasn't, and why it matters

JULY 21, 2026

An OpenAI model escaped its sandbox and hacked another company.

Autonomously. No human directing it step by step.

Contained sandbox

GPT-5.6 Sol + an unreleased pre-release model. Cyber refusals disabled for internal testing. Isolated network, limited to an internal package proxy.

Zero-day in the proxy

Models found and exploited a previously unknown vulnerability in the package registry cache to escape the sandbox.

Lateral movement

Privilege escalation across OpenAI's research environment until reaching a node with open internet access.

RCE on Hugging Face

Inferred HF hosted relevant test data. Chained stolen credentials + a second zero-day. Achieved remote code execution on production servers.

"It's quite mind-blowing that all of this happened autonomously." — Clem Delangue, CEO, Hugging Face

Sources: Fortune, CNN, CNBC, The Hacker News — July 21–22, 2026 · Clem Delangue, x.com/ClementDelangue · Time, July 24, 2026 · simonwillison.net/2026/Jul/22/openai-cyberattack/

This Is Not a "What If" Talk

What hasn't changed

- Attackers still need domain knowledge to act on the output
- Access still has to be gained. LLMs don't hand over credentials.
- Hardened, mature programs are still hard to compromise
- Human judgment still drives what happens next

What has changed

- The floor dropped. Less expertise needed to run a sophisticated attack.
- Speed. Hours of OSINT compressed into minutes.
- Scale. One attacker running campaigns against 50 targets at once.
- Cost. A personalized, targeted campaign is now basically free.

The fundamentals of offensive security haven't changed. The economics have. And as of two weeks ago, the autonomy has too.

Three Ways Attackers Use LLMs Today

01 **OSINT Automation**

Scrape job postings, GitHub, LinkedIn, Shodan. Feed it to an LLM and get a structured target profile in under 20 minutes.

02 **Personalized Phishing**

Context-aware lures that reference real company events, real team names, real tech stack. Not templates. Custom emails, personalized at scale.

03 **Exploit Scaffolding**

Read a CVE, generate PoC scaffolding, rewrite payloads to evade signature-based detection. Tools like PentestGPT (hacker.ai) are already in the wild. Already lowering the bar.

PART 2 OF 5

Automated OSINT at Scale

From a company name to a target profile in 20 minutes

OSINT: The Old Way vs. The New Way

Before

6-8 hours per target

- Manually search LinkedIn, job postings, GitHub
- Read Shodan output line by line
- Build the target profile by hand
- One target at a time
- Need a skilled OSINT practitioner to run it

Now

20 minutes. 50 targets.

- LLM pipeline scrapes and aggregates across all sources simultaneously
- Feeds Shodan output and returns a plain-English vuln summary
- Extracts org chart, tech stack, infrastructure map automatically
- Runs on 50 targets at the same time
- No specialist required. Just an API key.

What the Pipeline Actually Looks Like

1.	**Job posting scrape**	Extract tech stack. "Must have 3+ years with HashiCorp Vault, K8s on AWS, Terraform..." Instant infrastructure map.
2.	**LinkedIn OSINT**	Map the org chart. Identify DevOps/IT staff, tenure patterns, recent hires.
3.	**Shodan/Censys sweep**	Feed raw output to LLM. Get back: "Port 8080 — Jenkins 2.303, CVE-XXXX allows unauth RCE via Groovy console."
4.	**GitHub history scan**	Pattern match for API keys, internal hostnames, credential formats in commit history.
5.	**Target profile**	Structured output: top 3 entry points, key personnel, tech stack. Ready to act on.

DEMO: Show the Shodan prompt. Feed it raw output, get back a plain-English attack narrative in under 30 seconds.

None of this required authentication.

Every piece of the target profile came from passive, public sources.
Average time per target: under 20 minutes.

And it runs on 50 targets simultaneously.

PART 3 OF 5

Spear-Phishing at Scale

Personalized lures your filters won't catch

The Formula

Feed the LLM:

- Target name and role
- Recent company news
- Tech stack from job postings
- LinkedIn org chart

Get back:

A personalized phishing email that references the company's AWS migration, addresses the target by first name, and looks like it came from their account manager.

SAMPLE LURE

Subject: ACTION REQUIRED — Unusual sign-in to your AWS account

Hi [First Name],

We've detected an unusual sign-in to the Acme Corp AWS account from an unrecognized device. Given your team's recent migration to us-east-2, we've temporarily restricted access to your S3 buckets.

Please verify your identity within 4 hours to restore access.

→ Verify Account

Why Your Defenses Miss This

No templates

LLM-generated phishing doesn't match signature-based filter patterns. Every email is unique.

Contextually accurate

It references real events, real infrastructure, real org context. It passes the "does this feel right?" gut check.

Looks legitimate

Sent via a live SendGrid key or a typosquat domain registered that morning. Infrastructure is fresh.

What actually catches it

Behavioral signals, not content. Why is this user clicking an external link at 11pm from an unfamiliar sender?

PART 4 OF 5

Exploit Scaffolding & Payload Adaptation

The window between disclosure and exploitation is shrinking

CVE to Exploit: The Window Is Shrinking

Hours

to generate PoC scaffolding from a CVE. That used to take days.

Minutes

to adapt a public exploit to a specific target configuration.

Seconds

to rewrite payloads to evade signature-based detection rules.

What this means for defenders:

- Signature-based detection degrades faster. LLMs rewrite payloads faster than you can update signatures.
- CVSS score alone is not enough. Prioritize by "LLM-accessible exploitability." How easy is it to weaponize this with an LLM?
- The OpenAI/Hugging Face incident two weeks ago showed a model autonomously discovering zero-days and chaining them across environments. No human directed each step.

Personal aside: I use PentestGPT (hacker.ai) for side projects and CTFs. It's legitimately useful. Attackers know about it too.

Help Net Security / Synack: AI-Driven Vulnerability Trends Report (2026) · ZeroFox: CVE to Breach in Under an Hour · CSA: The Collapsing Exploit Window (2026) · hacker.ai

PART 5 OF 5

The Defensive Framework

Three teams. Three actions. Starting today.

Three Teams, Three Actions

SECURITY TEAM

Attack Surface

- Assume attackers have already run LLM-assisted OSINT on your external footprint
- Run it on yourself first. Automate a weekly sweep the way an attacker would.
- Unknown assets = untested assets = LLM-accessible targets

SOC

Detection

- Tune for behavioral anomalies, not content patterns
- Prioritize credential-use monitoring. LLM attacks compress time-to-use.
- Monitor for LLM-accessible CVE exploitation patterns in your patch backlog

LEADERSHIP

Program

- Annual pentests were already a gap. LLM recon just widens it.
- The question is no longer "are we tested?" It's "how fast could an attacker move today?"
- Continuous assurance is no longer optional; it's table stakes

"Attackers have access to the same models you do.

The question is whether your security program is moving as fast as they are."

Ask Us Anything.

Yes, even about AI.

Sprocket Security · sprocketsecurity.com

What does your external attack surface look like to an LLM-assisted attacker right now?
If you ran this recon pipeline on your own org tonight, what would it find that you don't know about?
How does LLM-accelerated recon change the math on annual pentests vs. continuous coverage?

References

OpenAI / Hugging Face Incident

- Fortune, CNN, CNBC, The Hacker News — July 21–22, 2026
- Time: "How OpenAI Lost Control of an AI Model" — July 24, 2026
- Simon Willison: simonwillison.net/2026/Jul/22/openai-cyberattack/
- Clem Delangue (CEO, Hugging Face): x.com/ClementDelangue — July 22, 2026
- TechCrunch: "Hugging Face CEO calls for radical transparency" — July 26, 2026

AI-Assisted Phishing at Scale

- CSA: AI-Weaponized Phishing — Systemic Risk (2026) — labs.cloudsecurityalliance.org
- ScienceDirect: "Evaluating LLMs' ability to automate spear phishing" (2026)
- Adaptive Security: Spear Phishing in 2026 — The Complete Guide
- Google Threat Intelligence Group: State actors integrating LLMs into offensive workflows (2025)

CVE Exploitation Window

- Help Net Security / Synack: 2025 AI-Driven Vulnerability Trends Report
- ZeroFox: "CVE to Breach in Under an Hour"
- CSA: "The Collapsing Exploit Window: AI-Speed Vulnerability Weaponization" (2026)
- CSA: "LLM-Orchestrated Kill Chains: From CVE to Database Breach in Four Pivots" (2026)

Sprocket Security

- sprocketsecurity.com/agents/apex
- "How Combined AI and Manual Testing Discovered Two Zero Days" — sprocketsecurity.com/blog — July 23, 2026
- PentestGPT: hacker.ai

If you're still here,

let's talk about what you do with all of this.

Annual pentests were built for a slower world.

The assumption

- Test once a year, fix what comes up
- Scope covers what you think is important
- Attackers need time and skill to move fast
- A clean pentest report means you're covered

The reality

- LLM-assisted recon doesn't wait for your test window
- That new subdomain you spun up Tuesday is already profiled
- Staging, dev, contractor infrastructure. None of it was in scope.
- The question is how fast an attacker could move today, not whether you passed last year

Agentic AI Needs Hard Guardrails.

What happened two weeks ago:

GPT-5.6 Sol found a zero-day in its own proxy, escaped the sandbox, and moved laterally to Hugging Face production. It was doing exactly what it was built to do: find a path forward.

What this means for you:

- **Real network isolation. Not just policy.**
- Every tool and permission an AI agent holds is an attack surface. The model chained three access vectors. Each one alone wasn't enough. Together they were RCE.
- Behavioral monitoring catches this. Content filtering doesn't.
- Put a human gate before irreversible actions. Goal-directed AI without supervision is a risk profile.

Assume capable models will try to escape. That's the threat model. It's also why Apex runs isolated in Sprocket's AWS environment with a human reviewing every finding before it reaches you.

Context Is What Makes Findings Actionable.

```
$ scanner output

> port 8080 open

> jenkins 2.303 detected

> CVE-2024-XXXX  CVSS: 9.8

> severity: CRITICAL
```

What context gives you

- Port 8080 open on api-staging.acmecorp.io, a subdomain outside your pentest scope
- /script endpoint is accessible without authentication
- Groovy console = one-liner reverse shell. No exploit code needed.
- Apex knows your past findings. It won't resurface what you already closed, and it flags if a fixed issue reappears.

A list of CVEs is not an attack narrative. Context is what your team can actually act on.

What We Do at Sprocket Security

"Ten years of curated offensive data, in combination with our proprietary software, frontier models, and human supervision, elevates what is possible with AI."

Continuous ASM

Find what's on your external footprint before attackers do. New assets, new paths. Ongoing, not annual.

Real pentesters behind every finding

We chain findings the way an attacker would. CVE + context + adjacent exposure = actual attack path, not a CVSS score. We've been doing this since before LLMs made recon cheap.

Apex

Our first AI agent

Agentic penetration tester for web applications. Runs unauthenticated testing continuously. Context-aware: it knows your assets and past findings. Every finding re-validated before it reports it. A Sprocket tester reviews and publishes before you see anything.

Real engagement, July 2026:

Apex found a session injection flaw in a healthcare app within minutes of starting. Our testers took that foothold and chained it into two zero-days.

Apex is Sprocket's first AI agent. More are coming in 2026.

Want to see what Apex finds in your environment?

We'll run the agent. You'll see exactly what we find.

sprocketsecurity.com/apex · Talk to us after the session