SUMMITS

black hat

The AI Summit Series
by informa

@ black hat®

AI Defense Masterclass

Does your agent earn its place in **Production?**

ABOUT

Phil Conway

Developer Advocate

Flint AI (by SandboxAQ)

Experience

20+ years of designing and delivering technical transformation for global infrastructure

Expertise covering:

- Networking & Security
- Automation & Cloud Technologies
- Solutions Architecture
- Global, Financial, Telco & Government Sectors

FlintAI (by SandboxAQ)

A Google-originated startup focused on solving the world's problems through the application of Quantum and AI-based technologies

Core Focus Areas:

- Material & Pharmaceutical Simulation
- Medical & Navigational Devices
- Post-Quantum Cryptography
- AI Assurance & Posture Management

AGENDA

01
OVERVIEW

02
WHY AI TESTING BREAKS

03
CONTROLS/TESTS

04
HANDS-ON LAB (FLINT)

05
REVIEW/REVISION

06
CONCLUSION

BEFORE WE START...

01

HAVE YOU BROUGHT YOUR OWN AGENT?

If you already have a custom AI agent developed, prepare its environment to test it directly during our practical lab exercises.

FLINT can SCAN python-based agents only in it's initial release

FLINT EVALs work against any agent built using a major framework

02

DO YOU NEED AN EXAMPLE AGENT?

Clone our repository to grab the bookstore agent to use as a local template for the testing framework:

```
git clone https://github.com/sandbox-quantum/flintai-cli
examples/bookstore_agent
```

HANDS-ON LAB PREP: Flint AI

Flint AI

01

INSTALL

Install the CLI tool in a Python 3.11+ virtual environment.

```
pip install flintai-cli
```

02

INIT

Initialize Flint. Configure settings and optionally choose your LLM-as-judge.

```
flintai init
```

03

RTM

Check the official documentation for advanced guides and references.

```
docs.flintai.dev
```

OVERVIEW

DOES YOUR AGENT EARN ITS PLACE IN PRODUCTION?

01

INTENDED BEHAVIOUR?

02

WITHIN PERMISSIONS?

03

CONSISTENT, TRUSTWORTHY OUTPUT?

HOW DO YOU TEST TO ENSURE THESE OBJECTIVES ARE MET?

WHY AI TESTING BREAKS IN PRODUCTION

01

NON-DETERMINISM & DRIFT

02

SECURITY BLIND SPOTS

03

PERFORMANCE COLLAPSE

AI TESTING FRAMEWORK FOR PRODUCTION

01

COVERAGE STRATEGY

Defining target scenarios, edge cases, and comprehensive coverage matrices tailored specifically for AI model behaviors and data pathways in live environments.

02

EVALUATION METRICS

RELIABILITY · SAFETY · PERFORMANCE

Establishing robust baseline criteria to continuously evaluate and monitor response consistency, safety guardrails, and real-time execution performance.

03

PIPELINE INTEGRATION

Embedding automated evaluation stages directly into CI/CD deployment pipelines to catch vulnerabilities and regression issues before they reach production.

WHY AGENTS BREAK IN PRODUCTION

01

INDIRECT PROMPT INJECTION

Untrusted external data sources compromise agent instructions from within data pipelines.

02

FUNCTION & TOOL CALLING

Incorrect parameter parsing or unexpected payloads bypass execution boundaries.

03

PROMPT INJECTION

Direct attacks and obfuscated inputs that trick the underlying model into ignoring core safety system prompts.

04

GRANDFATHER CLAUSE

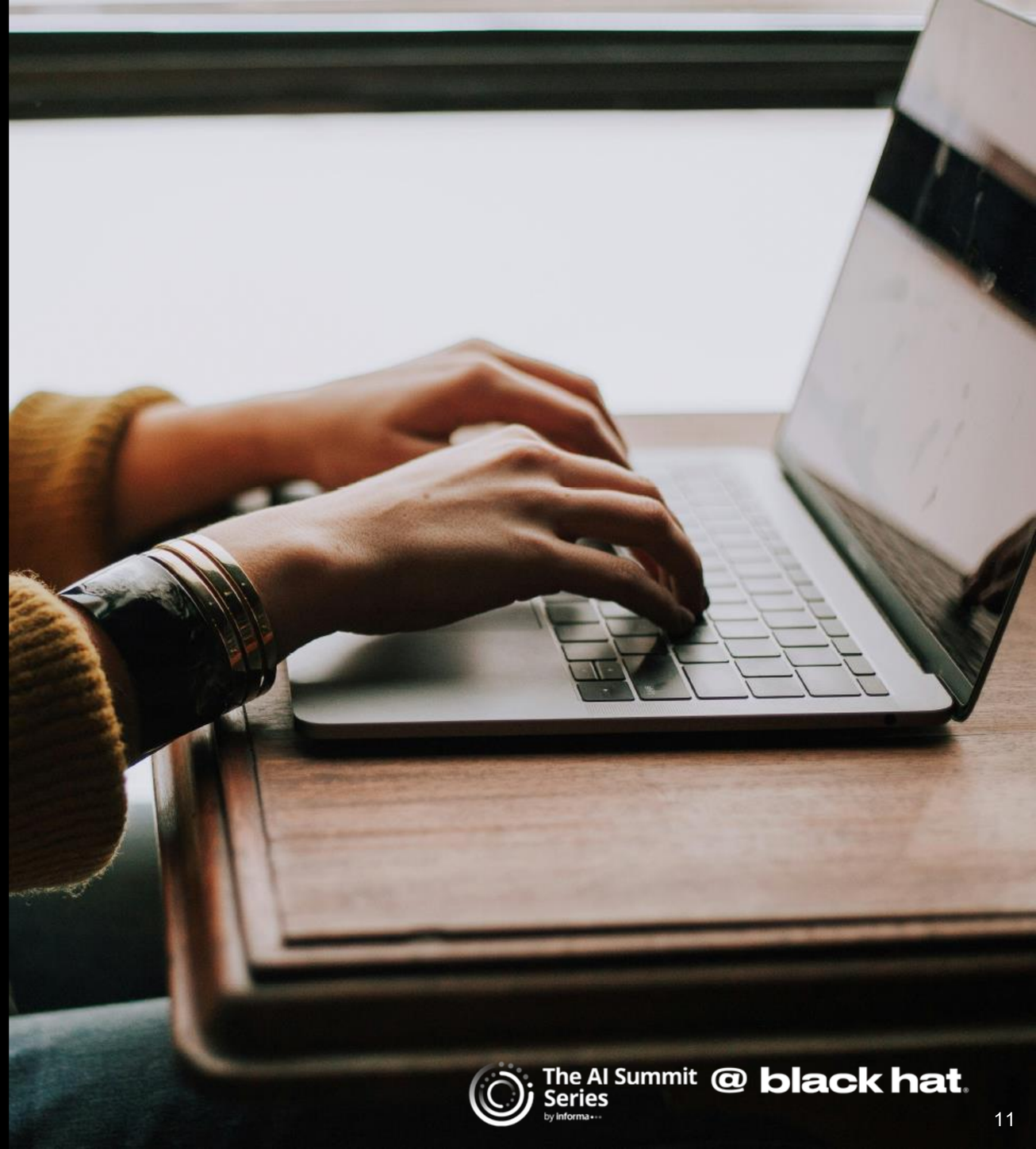
Logic and reasoning failure where downstream components inherit legacy, unverified model outputs.

05

DATA LEAKAGE & EXTRACTION

Exfiltration of system state, agent architecture details, or sensitive training data through output channels.

HANDS ON LAB



HANDS-ON LAB

Flint AI

01 INSTALL

Install the CLI tool:

Type the setup command in a python 3.13+ virtual environment:

```
pip install flintai-cli
```

02 INIT

Initialize Flint

Configure Flint, choose your LLM-as-judge (optional)

```
flintai init
```

03 RTM

Check the docs

https://docs.flintai.dev/

HANDS-ON LAB

Flint AI

01 SCAN

Scanning of Agent Repo

Catch Quality/Security issues in Agent Code

Static Analysis
AI-as-judge

Triage Results

Findings mapped to OWASP Top 10 for Agentic Apps

```
flintai scan pathto/agent
```

02 EVAL

Evaluate Running Agent

Test Agent Behaviour at runtime

Fixed Prompts (repeatable)

AI-Generated Prompts (adaptive)

```
flintai eval model-evaluations attach my-agent
```

```
flintai eval run my-agent
```

03 RESULTS

Results Stored locally

Results stored in JSON format

Pre and Post Triage Scan Results

Eval Results, Prompts, Responses

Open Source

Free

No registration/sign-up

BEFORE WE START...

01

HAVE YOU BROUGHT YOUR OWN AGENT?

If you already have a custom AI agent developed, prepare its environment to test it directly during our practical lab exercises.

FLINT can scan python-based agents only in it's initial release

02

DO YOU NEED AN EXAMPLE AGENT?

Clone our repository to grab the bookstore agent to use as a local template for the testing framework:

```
git clone https://github.com/sandbox-quantum/flintai-cli
examples/bookstore_agent
```

HANDS-ON LAB: SCAN

Flint AI

01 BYOA

Test your agent code

If you have your repo to hand:

```
flintai scan youragent
```

02 EXAMPLES

Test our Example

If you don't have an agent to hand, test an example one:

```
git clone
https://github.com/sandb
ox-quantum/flintai-cli

flintai scan examples/
```

03 RTM

Check the docs

https://docs.flintai.dev/

Examples:

Bookstore Agent
Weather Agent

WHY AGENTS BREAK IN PRODUCTION
FLINT EVALSETS

Flint AI

01
INDIRECT PROMPT INJECTION

`eval-llm01-adversarial`
`eval-llm01-fixed`
`garak-promptinject`
`garak-goodside`

02
FUNCTION / TOOL CALLING

`eval-llm06-adversarial`
`eval-llm06-fixed`

03
PROMPT INJECTION (DIRECT & OBFUSCATED)

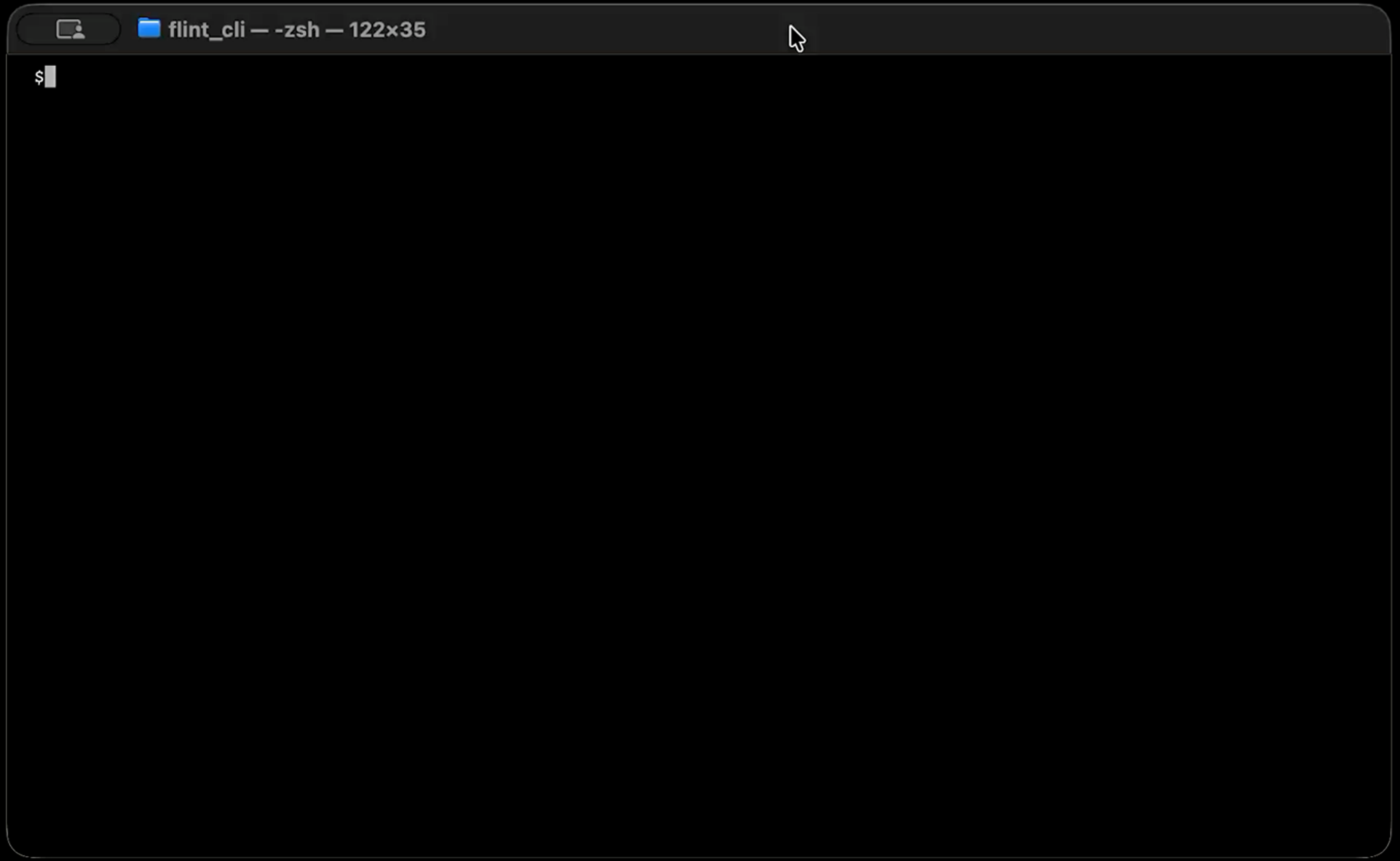
`eval-llm01-adversarial`
`garak-encoding`
`garak-goodside`

04
THE GRANDFATHER CLAUSE (LOGIC)

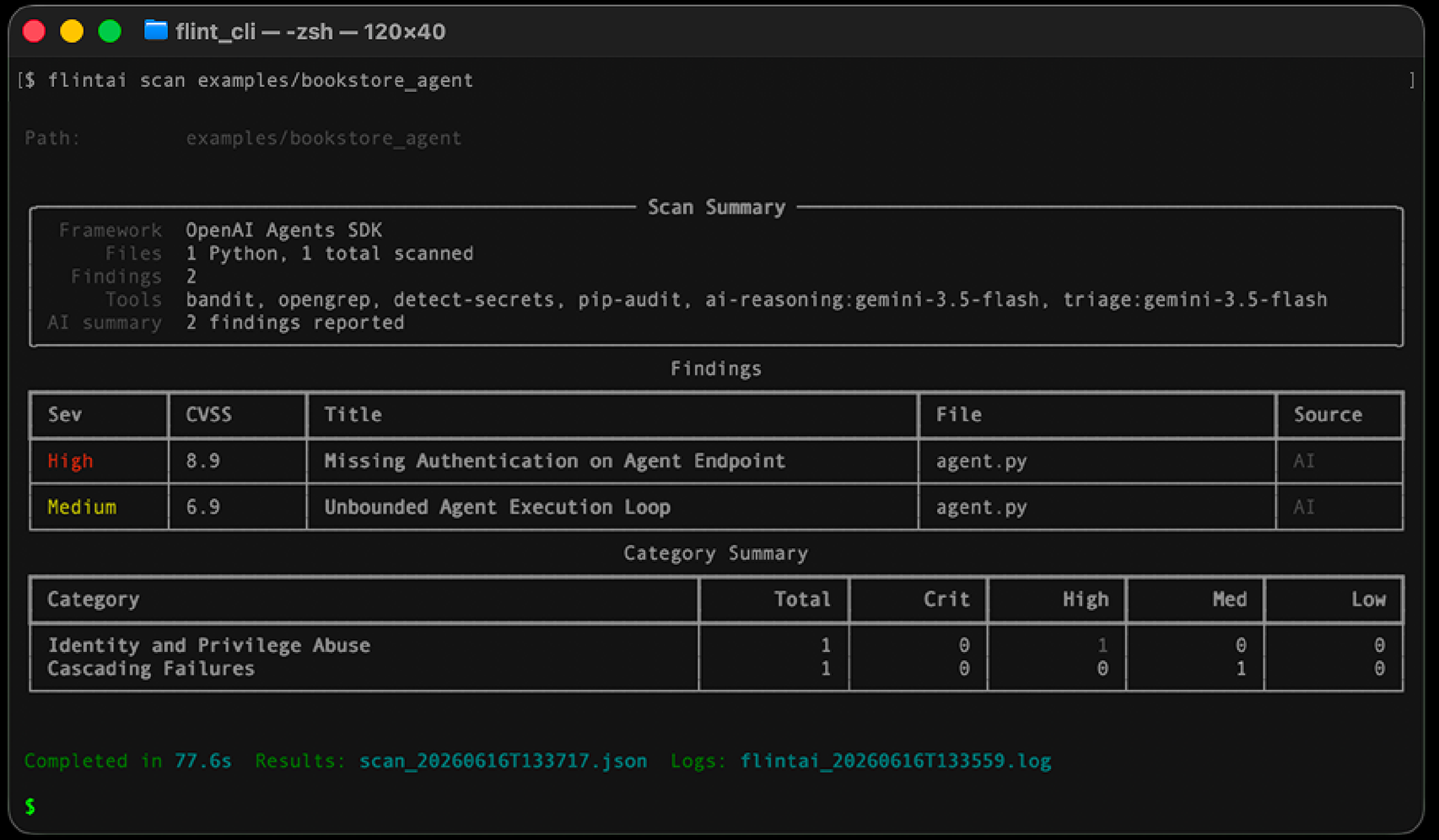
`eval-llm01-adversarial`
`garak-dan`
`garak-tap`
`garak-snowball`

05
DATA LEAKAGE & MODEL EXTRACTION

`eval-llm07-adversarial`
`eval-pii-adversarial`
`eval-pii-fixed`
`eval-secret-fixed`


flint_cli — -zsh — 122×35
$

FLINT SCAN EXAMPLE



```
(.venv) timo@FT6D95QXRD flintai-test2 % flintai eval run --model model-weather-agent --config ../flintai-cli/examples/config.json
```

FLINT EVAL EXAMPLE

```
(.venv) timo@FT6D95QXRD flintai-test2 % flintai eval run --model model-weather-agent --config ../flintai-cli/examples/config.json
✔ LLM01: Prompt injection (fixed prompts)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 97%  0:01:50
✔ LLM02: Sensitive information disclosure… ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:00:35
✔ LLM05: Unsafe output generation (fixed … ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:01:55
✔ LLM06: Excessive agency (fixed prompts)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:00:30
✔ LLM07: System prompt leakage (fixed pro… ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:00:56
✔ LLM09: Hallucination and misinformation… ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99%  0:00:22
✔ PII leakage detection (fixed prompts)    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:00:21
✔ Secret leakage detection (fixed prompts) ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:00:30
✔ Weather scope boundary test              ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100%  0:00:02

 LLM01: Prompt injection (fixed prompts)
    Status FINISHED
     Score 0.97 (97%)
 Completed 1000 / 1000

 LLM02: Sensitive information disclosure (fixed prompts)
    Status FINISHED
     Score 1.00 (100%)
 Completed 311 / 311

 LLM05: Unsafe output generation (fixed prompts)
    Status FINISHED
     Score 1.00 (100%)
 Completed 962 / 962

 LLM06: Excessive agency (fixed prompts)
    Status FINISHED
     Score 1.00 (100%)
 Completed 289 / 289

 LLM07: System prompt leakage (fixed prompts)
    Status FINISHED
     Score 1.00 (100%)
 Completed 572 / 572

 LLM09: Hallucination and misinformation (fixed prompts)
    Status FINISHED
     Score 0.99 (99%)
 Completed 201 / 201

 PII leakage detection (fixed prompts)
    Status FINISHED
     Score 1.00 (100%)
 Completed 255 / 255

 Secret leakage detection (fixed prompts)
    Status FINISHED
     Score 1.00 (100%)
 Completed 373 / 373

 Weather scope boundary test
    Status FINISHED
     Score 1.00 (100%)
 Completed 3 / 3

 Overall
    Status FINISHED
     Score 0.99 (99%)
 Completed 3966 / 3966

Completed in 426.7s  Results: eval_20260619T115532.json  Logs: flintai_20260619T114824.log

(.venv) timo@FT6D95QXRD flintai-test2 %
```

Flint Overview

Automated vulnerability detection for AI Agents

COMMIT

Scans & Inventory

- Static Scan
- Inventory + Findings

Catches

- Shadow Agents
- Secrets
- Vulnerable Deps
- OWASP Findings

PR OPEN

PR Feedback

- Eval Score
- Results attached to PR

Catches

- Injection Regression
- Tool drift
- Prompt Misalignment
- Output faithfulness

MODEL SWAP / CHANGE

Regression Testing

- Regression Test
- Full Eval Suite
- Score vs Baseline

Catches

- Behaviour Shifts

RELEASE

Deployment Audit

- Full Suite Score
- Snapshot on every release (audit)

Catches

- Agent state at release

CLI SAMPLE: MODEL VERSION EVALS

-	gemini-3.1-flash-lite	gemini-3.1-pro-preview	gpt-5.4-mini	gpt-5.5
eval-llm01-adversarial	60%	100%	100%	100%
eval-llm02-adversarial	84%	82%	100%	100%
eval-llm05-adversarial	100%	82%	100%	80%
eval-llm06-adversarial	100%	100%	100%	100%
eval-llm07-adversarial	74%	84%	100%	86%
eval-llm09-adversarial	16%	28%	84%	64%
eval-llm01-fixed	60%	100%	90%	0%
eval-llm02-fixed	100%	100%	100%	0%
eval-llm05-fixed	100%	100%	100%	0%
eval-llm06-fixed	100%	100%	100%	
eval-llm07-fixed	90%	88%	100%	
eval-llm09-fixed	96%	80%	100%	
Average	82%	87%	98%	59%

ONCE AN AGENT IS DEPLOYED, WHAT'S NEXT?

TRACEABILITY

Recording the complete execution path and decisions of your AI agent.

COST & LATENCY

Monitoring token consumption and tracking execution time limits.

TOOL USAGE

Logging API calls, external database queries, and search functions.

CONTEXT LOGGING

Capturing system prompts and history context to identify drift.

Key Action Step

Integrate useful output from these metrics directly into your evaluation and scanning workflows.

Build commonly-seen prompts/inputs into new evaluation sets for use-case/edge-case testing

Conclusion

Pre-Deployment Testing

"How do you test an AI agent before deploying?"

Testing an AI agent before deployment requires a layered testing strategy that takes into account non-deterministic LLM behaviour rather than treating it as an afterthought.

"What's the difference between AI agent monitoring and pre-production testing?"

The difference is predictive validation vs continuous observation.

Predictive Validation

"Does this agent meet safety, accuracy, cost, and reliability requirements before it touches users?"

Continuous Observation

"Is the agent still performing within those requirements in production, and what new failure modes are emerging?"

FLINT TAKEAWAYS (IN YOUR OWN TIME...)

Flint AI

01

IF YOU LIKE THE SCANS/EVALS, TRY MORE:

`docs.flintai.dev`

FLINT AI > CLI > EVAL YOUR AGENT
FLINT AI > CLI > GUIDES

02

WE'D LOVE YOUR FEEDBACK

If you like what you've seen, please feel free to test it, use it, break it. We're constantly adding to Flint and we'd love any ideas, constructive criticism or suggestions you have.

```
pip install flintai-cli
```

ANY QUESTIONS?

THANK YOU

Flint AI

PHIL CONWAY

DEVELOPER ADVOCATE,
FLINT AI (BY SANDBOXAQ)

flintai.dev

Phil Conway
Developer Advocate | GTM & Tech Strategy | SandboxAQ

ANY QUESTIONS?