

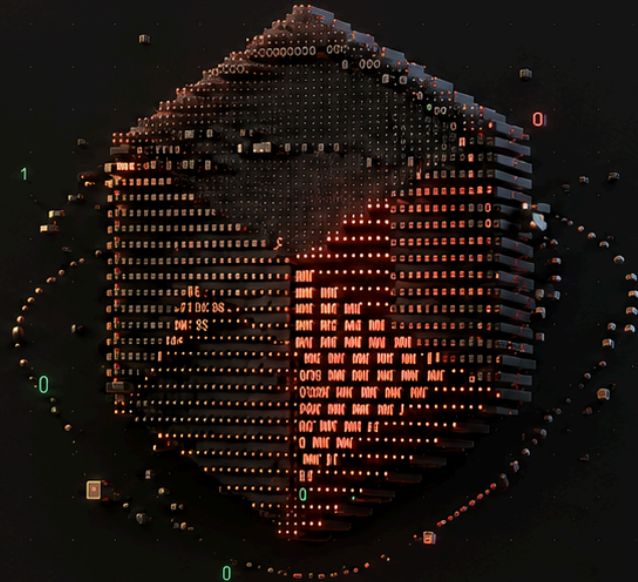


VIGOLIUM

// What is Vigolium?

# SERIOUS SECURITY AUDIT, NOT JUST A PR REVIEW

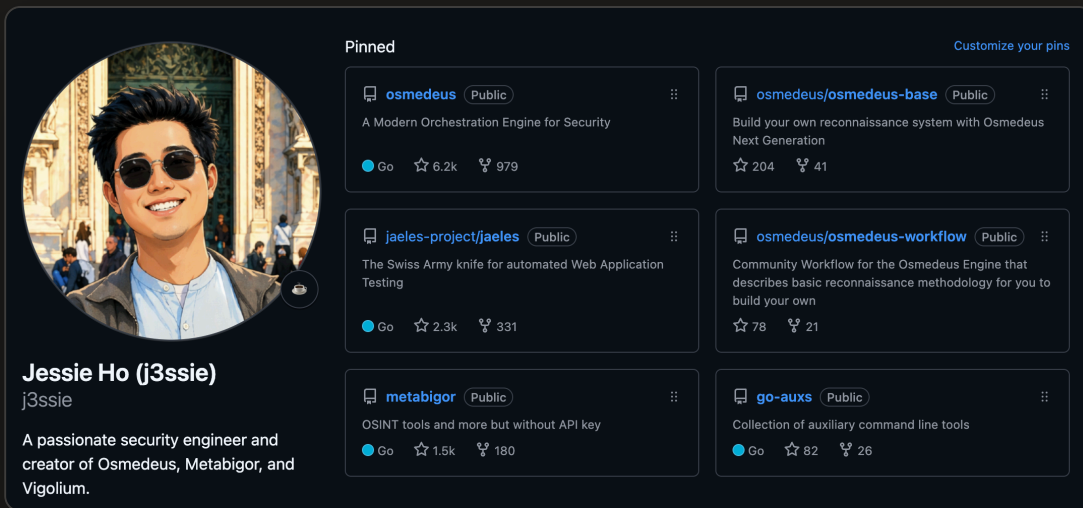
Catch critical vulnerabilities that traditional scanners and AI code review tools miss, with validated proof your team can act on.



[www.vigolium.com/share/technical](http://www.vigolium.com/share/technical) / [docs.vigolium.com](http://docs.vigolium.com) / [jessie@vigolium.com](mailto:jessie@vigolium.com)

# Whoami

- Known as **j3ssie** – Senior Security Engineer at TikTok
- **Open-source lover** building security tools for practitioners.
- **Recognized by Fortune 500 leaders** including Google, Apple, Microsoft, Netflix, Snapchat, the US Department of Defense, and many more for identifying security vulnerabilities in their products.



**Jessie Ho (j3ssie)**  
j3ssie  
A passionate security engineer and creator of Osmedeus, Metabigor, and Vigolium.

**Pinned** Customize your pins

Repository	Stars	Forks
<a href="#">osmedeus</a> (Public)	6.2k	979
<a href="#">osmedeus/osmedeus-base</a> (Public)	204	41
<a href="#">jaeles-project/jaeles</a> (Public)	2.3k	331
<a href="#">osmedeus/osmedeus-workflow</a> (Public)	78	21
<a href="#">metabigor</a> (Public)	1.5k	180
<a href="#">go-auxs</a> (Public)	82	26

# The gap

---

Continuous delivery never got continuous security testing

- **Manual pentests** ship as a PDF every 6-12 months. Stale before the next sprint, and never re-run on the diff
- **Traditional DAST/SAST** scan from a fixed checklist. No application context, no exploit synthesis, and false-positive rates that demand a full-time triage owner
- **Diff-scoped AI reviewers** read the patch, not the running app. They emit prose, never an HTTP request
- **Vibe-coded apps** are growing fast. More production software is being generated quickly, often without matching security review depth

Nothing in the standard stack ingests the **whole repo**, **plans an attack**, **issues live HTTP probes**, and **validates the result** end-to-end. That's the slot Vigolium fills.

# AI code review is not security testing

A diff reviewer reasons over a patch. A security scanner attacks a running system. Different inputs, different outputs, different jobs.

## Scope

**AI code reviewers** see the lines in the PR plus a small retrieved context window.

**Vigolium** ingests the entire repo – every route, auth flow, and downstream service – and binds it to the live target.

*A cross-endpoint auth bypass is not visible from one file.*

## Action

**AI code reviewers** reason about source and emit suggestions.

**Vigolium** synthesises payloads, issues live HTTP requests against the target, and reads the response.

*Static reasoning cannot prove exploitability. A confirmed request/response pair can.*

## Signal

**AI code reviewers** produce unvalidated prose.

**Vigolium** produces a finding with the request, response, and a reproduction command.

*Diff annotations are hints. HTTP records are evidence.*

# Architecture

3 components. The CLI does all the work; Workbench are different ways to view it.

## Vigolium CLI

Single Go binary. Hosts the **native scanner** (235+ modules), the **olium runtime** (LLM dispatch), and the **agent subcommands**. Also runs as a REST API server.

## Vigolium Workbench

Self-hosted dashboard. Reads the same DB store the CLI writes to.

## Static Report

Portable HTML output with request/response evidence for offline review and sharing generated from the CLI results.



**The goal: build systems that don't become obsolete.**  
Give agents the best native tools, and every new model release makes the system stronger.

# Scanning modes

Two top-level commands. Both write to the same store, both produce the same finding/HTTP-record schema.

```
vigolium scan
```

**Deterministic.** Native Go pipeline, no LLM in the loop.

- 6 phases: heuristics → external harvest → discovery → spidering → known-issue-scan → dynamic-assessment
- 235 modules (144 active, 91 passive)
- Three strategies: `lite` / `balanced` (default) / `deep`
- Repeatable, fast, CI-friendly

**Best for: large amount of application, baseline coverage.**

```
vigolium agent <mode>
```

**AI-driven.** All dispatch through the in-process olium runtime ( `pkg/olium/` ).

- 8 subcommands: `query`, `autopilot`, `swarm`, `archon`, `piolium`, `audit`, `olium`, `session`
- Source-aware via `--source` (clones git URLs or reads local paths)
- Provider selection in config: `claude-oauth`, `codex-oauth`, `anthropic-api-key`, `openai-api-key`, `claude-code-cli`

# Native scan – `vigolium scan`

## Pipeline (6 phases)

1. Heuristics Pre-flight: WAF, redirects, tech fingerprint
2. External Harvest Wayback, CommonCrawl, OTX, URLScan, VirusTotal
3. Discovery Brute-force dirs/files, JS analysis
4. Spidering Chromium SPA crawler, form filling
5. KnownIssueScan Nuclei templates + Kingfisher secrets
6. Dynamic-Assess 144 active + 91 passive modules

## Strategies

Phase	lite	balanced	deep
External harvest			yes
Discovery		yes	yes
Spidering		yes	yes
Known-issue		yes	yes
Dynamic-assess	yes	yes	yes

## Module dispatch (scan scope)

Three granularities, declared per module as a bitmask:

- `ScanScopeRequest` – once per request (403 bypass, JWT, cache poison)
- `ScanScopeInsertionPoint` – once per parameter (SQLi, SSTI, XSS)

## Engine internals

- Worker pool with per-host rate limiting + hybrid queue
- **Value-aware mutation** – classify parameter values by semantic type, generate type-specific payloads
- OAST callbacks for blind detection (blind SSRF, blind XXE, blind RCE)
- Browser-driven SPA spider with form filling + JS analysis

**Severity scale:** critical > high > medium > low > suspect > info. **Confidence scale:** certain (payload executed) > firm (behavioral) > tentative (heuristic). Every finding carries both.

# Inputs & integrations

What you point Vigolium at, how it authenticates, where it plugs in.

## Input formats

- **URL / target list** – `-t https://example.com`, file, or stdin
- **OpenAPI / Swagger** – JSON or YAML
- **Postman** – collection export
- **Burp Suite** – XML state file and Burp Suite Plugin
- **cURL** – raw `curl ...` command
- **Raw HTTP** – request/response file

```
GET /catalog?searchTerm=apple HTTP/2
Host: localhost:3000
...
```

## Source code (for agentic modes)

```
--source <local-path | git-url> - canonical flag across
swarm, autopilot, archon, piolium, audit, query.
```

Git URLs are cloned into the session dir.

Host	Method	Path	Status	Content_Type	Size	Words	Title	Source
http://localhost:3000	POST	/api/feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	GET	/api/socket.io/	200	text/html	2	20		finding
http://localhost:3000	POST	/socket.io/?EIO=4&transport=polling&...	200	text/html	2	20		finding
http://localhost:3000	POST	/socket.io/?EIO=4&transport=websocket&...	200	text/html	2	20		finding
http://localhost:3000	POST	/api/feedbacks/	200	text/html; charset=utf-8	0	45		finding
http://localhost:3000	POST	/api/feedbacks/	200	text/html; charset=utf-8	0	45		finding
http://localhost:3000	PUT	HTTP/1.1	200	text/html; charset=utf-8	0	45		finding
http://localhost:3000	POST	HTTP/1.1	200	text/html; charset=utf-8	0	45		finding
http://localhost:3000	GET	/chunk-BBQJZW7H.js	200	application/javascript...	236653	5862		deparos
http://localhost:3000	GET	/scripts	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/styles	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/favicon.js	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/assets/public	302	text/html; charset=utf-8	167	49	Redirecting	deparos
http://localhost:3000	GET	/main	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/ftp	200	text/html; charset=utf-8	11296	969	Listing directory /ftp	deparos
http://localhost:3000	GET	/public	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/favicon.js?html	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/ftp/juice-shop	403	text/html; charset=utf-8	1935	108	Error: only file and json fl...	deparos
http://localhost:3000	GET	/ftp	200	text/html; charset=utf-8	11253	971	Listing directory /ftp	deparos
http://localhost:3000	GET	/assets/favicon.js	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/assets/public	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/assets/scripts	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/scripts	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/assets/styles	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/assets/polyfills	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/favicon.js	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	/styles	200	text/html; charset=utf-8	75802	3650	OWASP Juice Shop	deparos
http://localhost:3000	GET	HTTP/1.1	200	text/html; charset=utf-8	0	45		scanner
http://localhost:3000	POST	/api/feedbacks/	401	text/html; charset=utf-8	42	38		spidering
http://localhost:3000	POST	/api/feedbacks/	401	text/html; charset=utf-8	42	38		spidering
http://localhost:3000	GET	/chunk-BBQJZW7H.js	304	application/javascript...	236653	5859		spidering
http://localhost:3000	POST	/rest/user/login	401	text/html; charset=utf-8	26	35		spidering
http://localhost:3000	GET	/chunk-BBQJZW7H.js	200	application/javascript...	236653	5864		spidering

# Sessions & JS extensions

## Sessions – `--auth-file`

Multi-session YAML/JSON. `primary` drives discovery;

`compare` sessions are tested against it for IDOR.

```
# idor-test.yaml
sessions:
- name: admin
  role: primary
  headers:
    Cookie: "${ADMIN_SESSION_COOKIE}"

- name: regular_user
  role: compare
  headers:
    Cookie: "${USER_SESSION_COOKIE}"

# Login flow with token extraction
- name: api_admin
  role: primary
  login:
    url: https://app/api/auth/login
    method: POST
    body: '{"u":"admin","p":"admin123"}'
    type: bearer
    token_path: .data.access_token
```

## Embedded Javascript Runtime

JS extensions implements active or passive module shape.

APIs: `vigolium.http`, `.db`, `.parse`, `.utils`, `.agent`.

```
// webhook_callback_detector.js – passive
module.exports = {
  id: "webhook-callback-detector",
  name: "Webhook/Callback URL Detector",
  type: "passive",
  severity: "suspect",
  confidence: "tentative",
  scope: "response",
  scanTypes: ["per_request"],

  scanPerRequest: function(ctx) {
    if (!ctx.response || !ctx.response.body) return null;
    var ct = ctx.response.headers["content-type"] || "";
    if (ct.toLowerCase().indexOf("json") === -1) return null;

    var hits = vigolium.utils.regexExtract(
      ctx.response.body,
      "(?:callback_url|webhook|notify_url)" + "\\s*:\\s*" + "(https?://[^\s]+)";
    );
    if (!hits || !hits.length) return null;

    return [{
      url: ctx.request.url,
      name: "Webhook/Callback URL exposed",
      matched: hits.join(", ")
    }];
  }
};
```

# Coverage

---

OWASP Top 10 and beyond – broken out by the categories the scanner registry actually uses.

## Injection

XSS (reflected, DOM-based, SSR hydration) · SQLi (error-based, boolean/time-blind) · NoSQLi (Mongo/Couch/Cassandra) · SSTI/CSTI · CRLF · OS command exec · XXE · prototype pollution · insecure deserialization

## Access control

CSRF · IDOR · authorization bypass · mass assignment · 403/401 forbidden bypass · HTTP method tampering

## File & path

LFI · path traversal (null byte, double encode, Unicode) · file upload flaws · directory listing · backup discovery

## Framework-specific

Spring Boot (actuators) · Django · Laravel · Rails · Express · Next.js · Nuxt · Remix · ASP.NET / Blazor · Flask · FastAPI

## CMS

WordPress (XML-RPC, user enum, AJAX) · Drupal · Joomla · CMS installer exposure

## Cloud & infra

Firebase (RTDB, storage, auth, functions) · cloud storage listing/takeover · default credentials · web cache poisoning · CORS misconfig · host header injection

# Agentic modes – `vigolium agent <mode>`

8 subcommands, four families. All dispatched through the in-process `olium` runtime.

Subcommand	Family	Job
<code>autopilot</code>	agentic scan	One long LLM session with full Bash/Read/Write/Grep/Glob until <code>halt_scan</code> . Free-form pentest.
<code>swarm</code>	agentic scan	10-phase pipeline. Master agent picks modules, generates JS extensions; native engine executes; optional triage + rescan loop.
<code>archon</code>	source audit	Multi-phase whitebox audit on Claude/Codex/OpenCode. Modes: <code>lite</code> (3) · <code>balanced</code> (6) · <code>deep</code> (10).
<code>piolium</code>	source audit	Same audit shape, Pi runtime. Useful for non-Anthropic providers (OpenAI/Codex, Gemini).
<code>audit</code>	source audit	Unified driver – runs <code>archon</code> + <code>piolium</code> back-to-back, dedups findings under one parent scan.
<code>olium</code> / <code>ol</code>	TUI	Interactive multi-turn chat. Provider-agnostic. Not orchestrated scanning.

`olium runtime providers` ( `agent.olium.provider` ) `codex-`  
`oauth` · `anthropic-api-key` · `openai-api-key` · `claude-`

`Session artifacts` at `~/.vigolium/agent-sessions/` Output  
schemas: `findings` , `http_records` , `attack_plan` ,

# Deep audit – 15 phases

```
vigolium agent archon --mode deep – full whitebox security audit. Swarm-orchestrated, capped at 3 concurrent agents.
```

#	Phase	One-liner	#	Phase	One-liner
1	<b>Intelligence Gathering</b>	Pull CVE/GHSA advisories + mine commit history for security-relevant changes.	9	<b>Spec Gap Analysis</b>	Diff documented behavior vs. implementation; find undocumented features.
2	<b>Patch Bypass Analysis</b>	Re-derive every fixed CVE / suspicious patch and try to bypass it.	10	<b>Review Chambers</b>	Synthesizer + Ideator + Tracer + Devil's Advocate debate every cluster.
3	<b>Knowledge Base</b>	Build DFD/CFD slices, trust boundaries, attack surface map.	11	<b>FP Elimination</b>	fp-check sweep, then cold-context verifier on every CRITICAL/HIGH.
4	<b>Static Analysis</b>	CodeQL + Semgrep with inline security-relevance enrichment.	12	<b>Variant Analysis</b>	Variant-hunter per confirmed bug – find the same shape elsewhere.
5	<b>Deep Probe</b>	Per-component teams: strategist, backward + contradiction reasoners, evidence harvester.	13	<b>PoC Construction</b>	Working exploit + evidence per finding; consolidate into stable IDs.
6	<b>Authorization Audit</b>	Enumerate every route/handler; build the authz matrix.	14	<b>Finding Finalization</b>	Author per-finding <code>report.md</code> from cold context – gate before assembly.
7	<b>State &amp; Concurrency</b>	Catalogue stateful entities + concurrency primitives; hunt races.	15	<b>Final Report</b>	Assemble <code>final-audit-report.md</code> from per-finding reports.
8	<b>Cross-Service Taint</b>	Propagate taint across service boundaries (queue, RPC, shared DB).			





# Competitive landscape

Where Vigolium sits relative to the existing AppSec stack.

Capability	Pentester	Scanner (DAST/SAST)	Bug bounty	AI code reviewer	Vigolium
Whole-repo source context	⚠️	✖️	✖️	✖️	✓
Runtime exploitation (DAST)	✓	⚠️	✓	✖️	✓
Validated PoC (request/response)	✓	⚠️	✓	✖️	✓
AI triage / FP filtering	✖️	✖️	✓ (human)	✖️	✓
AI-driven payload synthesis	✓ (human)	✖️	✓ (human)	✖️	✓

Vigolium combines **SAST** (whole-repo source context) + **DAST** (live exploitation) + **AI triage** + **agentic autopilot** in one binary.

# Validation methodology

## Reproducible evidence

Each finding ships with the **HTTP request**, **response**, **module ID**, **payload**, and **confidence rating** replayable from the CLI or Workbench.

## Open-source bug bounties

Continuous runs against in-the-wild OSS projects. Findings disclosed to maintainers. Recall measured against issues those projects have already triaged.

## Benchmark suite

Adversarial verdicts + cold verification – agent’s deep mode runs adversarial debate chambers and a cold-verify pass before a finding is written.

The screenshot displays the Vigolium Static Report interface for the target 'ollama'. The interface includes a search bar, filters for severities, confidence, modules, sources, and tags, and a '45 of 45 findings' indicator. A table lists findings with columns for ID, severity, module, and description. Finding #9 is highlighted as a HIGH severity issue. A detailed view of finding #9 is shown on the right, including a summary, a list of three compounding defects, and a detailed description of the defects. The defects are: 1. Pipe-only splitting: extractBashPrefix (X/agent/approval.go:284-286) only splits on | before extracting the prefix. Any ;, &&, ||, \$(...), backticks, or newlines fall inside parts[] – they are never examined. 2. Denylist as substring: isDenied (X/agent/approval.go:94-122), called from X/cmd/run.go:378 uses strings.Contains(cmd, pattern) with literal patterns (cmd, 'rm -rf', 'curl -X POST', '\*/\*/shadow', etc.). Any inline empty-quote (cmd, '\*/\*/shadow', etc.) breaks the match while bash still evaluates the intended command. 3. bash -c executor: X/tools/bash.go:64 invokes exec.CommandContext(ctx, "bash", "-c", cmd). Bash is invoked AS a shell – metachars are interpreted by /bin/bash, not by Go's execve. This is the sole execution sink for approved commands. The cache key produced by extractBashPrefix ignores everything after the first path-like argument. Once a user approves (cat /tools/run.go) for the session, the prefix (cat/tools) enters the allowlist, and the agent's subsequent tool-calls reach IsAllowed=true when their first pipe-segment's first non-flag path arg happens to start with /tools/ – no matter what metachars follow. Details: Three compounding defects produce a single command-injection primitive in the agent's approval layer: 1. Pipe-only splitting: extractBashPrefix (X/agent/approval.go:284-286) only splits on | before extracting the prefix. Any ;, &&, ||, \$(...), backticks, or newlines fall inside parts[] – they are never examined. 2. Denylist as substring: isDenied (X/agent/approval.go:94-122), called from X/cmd/run.go:378 uses strings.Contains(cmd, pattern) with literal patterns (cmd, 'rm -rf', 'curl -X POST', '\*/\*/shadow', etc.). Any inline empty-quote (cmd, '\*/\*/shadow', etc.) breaks the match while bash still evaluates the intended command. 3. bash -c executor: X/tools/bash.go:64 invokes exec.CommandContext(ctx, "bash", "-c", cmd). Bash is invoked AS a shell – metachars are

# OSS audit corpus

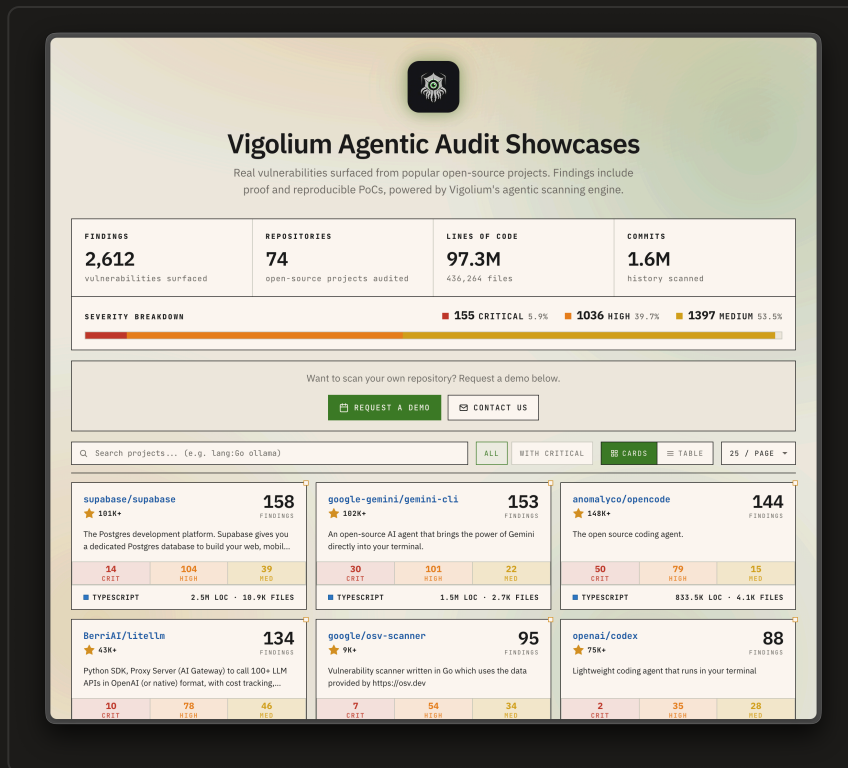
Vigolium pointed at popular open-source projects – code already running inside Fortune 500 stacks, already scanned by traditional tools.

## Corpus

Projects scanned	46
Files analysed	263,406
Lines of code	52,902,830
Commits ingested	931,160
Findings surfaced	1,113

## Severity

Severity	Count
Critical	16
High	323



# Anatomy of a scan

What happens between `vigolium scan -t https://app` and a finding written to the store.

```
CLI invocation (cmd/vigolium/main.go)
| flag parse, config load, DB init
▼
Input parsing (pkg/input/source/)
| URL/file/stdin → WorkItem stream
▼
Runner orchestration (internal/runner/)
| 6-phase pipeline, strategy-driven
▼
Executor (pkg/core/executor.go)
| worker pool, hybrid queue,
| per-host rate limit
▼
Module dispatch (pkg/modules/)
| passive (sequential) +
| active (parallel),
| scoped per host/req/insertion-point
▼
Result emission (pkg/output/output.go)
  post-hooks → SaveFinding →
  OnResult → Notify
```

## State

SQLite by default; Postgres for multi-tenant

Workbench/Console. Schema covers `targets`, `findings`, `http_records`, `agentic_scans`, `oast_interactions`, `extensions`.

## Where agentic plugs in

`swarm` runs the same pipeline but injects an **AI plan phase** before scan and an optional **AI triage phase** after. Plan picks modules + writes JS extensions; triage requests targeted re-scans on disputed findings.

# Deployment models

Same binary, same finding schema, three operating shapes. Pick by data-residency and team size.

## CLI-only

Single Go binary, local SQLite.

For: solo operators, CI runners, ad-hoc audits.

- `vigolium scan -t URL`
- Outputs: console, JSONL, HTML report, SARIF
- BYO LLM credentials for `agent` modes
- Zero infra; data never leaves the box

## Self-hosted Workbench

CLI + dashboard in your VPC.

For: teams who need a UI but can't ship data outside their boundary.

- Same CLI engine, Postgres backend
- Multi-tenant projects + RBAC
- Browse findings, replay HTTP records
- Bring-your-own LLM key or local Codex/Claude

## Vigolium Console

Managed cloud, hosted by us.

For: teams who'd rather not run infra.

- Same UI as Workbench
- Scheduled scans, GitHub/GitLab webhooks
- Org/team management, SSO
- Scan compute managed; LLM keys pooled

# Demo

The screenshot displays the 'FINDINGS' tab of the Vigolium scanner. It shows a table of 25 findings with columns for ID, SEV, COMP, ISSUE, TYPE, DESCRIPTION, and HTTPD\_AT. The findings include various vulnerabilities such as missing security headers, session cookie issues, and endpoint anomalies.

ID	SEV	COMP	ISSUE	TYPE	DESCRIPTION	HTTPD_AT
11	INFO	Firm	info	passive	Missing security header...	https://log...
2	HIGH	Firm	ssx	active	Reflected XSS via 'q'...	https://log...
26	SUSPECT	tentative	cookie-reflect	passive	Session cookie value...	https://mm...
20	SUSPECT	tentative	ext-external-ptr	active	XML external entity re...	https://mm...
24	SUSPECT	Firm	method-override	active	X-HTTP-Method-Override...	https://ml...
23	SUSPECT	Firm	encoding-bypass	active	Double URL-encoded path...	https://ml...
22	SUSPECT	Firm	header-reflect	active	Header-controlled input...	https://mm...
21	SUSPECT	tentative	content-type-x	passive	Endpoint normally retu...	https://ml...
28	SUSPECT	Firm	timing-anomaly	active	Response time for 'hml...	https://log...
19	SUSPECT	tentative	response-anom	passive	Error response body sh...	https://mm...
18	SUSPECT	Firm	wildcard-inject	active	Wildcard character 'x'...	https://ml...
17	SUSPECT	tentative	anomaly-ranking	passive	Response is a statistic...	https://ml...
16	SUSPECT	tentative	input-behavior	active	HTML tag count change...	https://mm...
15	SUSPECT	Firm	enum-behavior	active	Differential behavior...	https://ml...
14	SUSPECT	Firm	suspect-transf	active	Expression evaluated L...	https://ml...
13	SUSPECT	Firm	backslash-tran	active	Backslash consumed in...	https://mm...
10	LOW	Firm	info	passive	Server version disclos...	https://log...
9	MEDIUM	Firm	openredirect	active	Open redirect via 'url...	https://log...
6	HIGH	certain	lfi	active	Local file inclusion v...	https://log...
7	HIGH	certain	ssll	active	Server-side template L...	https://ml...
8	MEDIUM	Firm	cveid	active	CVE injection in 'loc...	https://ml...
5	CRITICAL	certain	lfi	active	Local file inclusion v...	https://ml...
1	HIGH	Firm	ssx	active	Reflected XSS via 'q'...	https://mm...
12	MEDIUM	Firm	info	passive	API response includes...	https://ml...
4	HIGH	certain	ssll	active	Error-based SQL inject...	https://ml...
3	HIGH	certain	ssll	active	Header-based SQL inject...	https://ml...

The screenshot displays the 'HTTP RECORDS' tab of the Vigolium scanner. It shows a table of HTTP requests and responses with columns for ID, SEV, SC, HOST, PATH, TYPE, SIZE, and TITLE. The records include various requests such as OPTIONS, GET, and POST to different endpoints like /api-shop/sec and /api-shop/loc.

ID	SEV	SC	HOST	PATH	TYPE	SIZE	TITLE	USER
403	OPTIO	403	api-shop.sec	/api/v3/admin/config	text/plain	52	0	
208	GET	208	localhost	/HTTP/1.1	text/html	73.2	0	OWASP Juice Shop
208	POST	208	example.com	/api/shop/userServic...	text/xml	399	0	
208	POST	208	example.com	/graphql	application/...	122	0	
161	GET	161	example.com	/api/notifications	text/html	0	0	
208	GET	208	example.com	/account/preferences	text/html	716	0	Preferences
208	GET	208	api-shop.sec	/api/v3/admin/report...	application/...	81	0	
413	GET	413	api-shop.sec	/api/v3/api/load	application/...	105	0	
504	GET	504	api-shop.sec	/api/v3/allow-endpoi...	text/html	152	0	504 Gateway Time...
208	GET	208	admin.example.com	/admin/report/fore...	text/csv	102	0	
208	GET	208	admin.example.com	/admin/report/fore...	text/plain	209	0	
208	POST	208	admin.example.com	/admin/settings	text/html	376	0	Settings Saved
208	GET	208	admin.example.com	/admin/settings	text/html	626	0	Settings - Admin
208	GET	208	admin.example.com	/admin	text/html	398	0	Admin Panel
302	GET	302	legacy.example.com	/redirect?url=https...	text/html	0	0	
301	GET	301	legacy.example.com	/old-page	text/html	0	0	
208	POST	208	legacy.example.com	/api/v3/admin/rel	text/html	222	0	Form Submitted
208	GET	208	legacy.example.com	/page.php?page=...	text/html	452	0	Legacy Portal
208	GET	208	legacy.example.com	/	text/html	456	0	Legacy Portal
208	GET	208	blog.test	/feed/rss	applicatio...	883	0	
208	GET	208	blog.test	/search/page=...	text/html	284	0	Search Results
208	GET	208	blog.test	/tags/security	text/html	384	0	Posts Tagged 'Sec...
201	POST	201	blog.test	/post/hello-world/c...	text/html	188	0	
208	GET	208	blog.test	/post/api/objectio...	text/html	104	0	500: Discontin...
208	GET	208	blog.test	/post/hello-world	text/html	1.2	0	Hello World - BL...



Pre-recorded results: <https://demo.vigolium.com/>  
Reach out to: [jessie@vigolium.com](mailto:jessie@vigolium.com)

# Agentic audit

```
codex main
j3ssie ► vigolium agent audit --driver archon --source _ --agent claude
+ Vigolium Audit
  ◆ Driver: archon | Mode: balanced
  ◎ Source: /Users/j3ssie/Desktop/demo/codex
  ◆ Session: /Users/j3ssie/.vigolium/agent-sessions/9cdca776-b6d9-481e-b23b-0b6352f0a941

· → archon (/Users/j3ssie/.vigolium/agent-sessions/9cdca776-b6d9-481e-b23b-0b6352f0a941/archon)
· Archon preflight check... ok platform=claude reply="Ready." in 6s

[stream] session 30019579 model=claude-opus-4-7[1m] cwd=/Users/j3ssie/Desktop/demo/codex
→ Skill ("skill":"archon-audit:archon:balanced")
← Launching skill: archon-audit:archon:balanced
→ Bash ("command":"git rev-parse --is-inside-work-tree 2\u003e\u00261 \u0026\u0026 git rev-parse HEAD \u0026\u0026
6 git branch --show-current \u0026\u0026 pwd \u0026\u0026 echo \"ARCHON_REP...")
→ Bash ("command":"ls -la /Users/j3ssie/Desktop/demo/codex/ 2\u003e\u00261 | head -40","description":"List repo c
ontents")
→ Bash ("command":"test -f /Users/j3ssie/Desktop/demo/codex/archon/INFO.md \u0026\u0026 echo \"INFO.md exists\" |
| echo \"no INFO.md\"","description":"Check for INFO.md")
← true 2004173cd761b9d6a8ada9c8597dd93b22b4b5a3 main /Users/j3ssie/Desktop/demo/codex ARCHON_REPOSITORY=openai/co
dex ARCHON_INFO_AVAILABLE=false
← no INFO.md
← total 3512 drwxr-xr-x@ 47 j3ssie staff 1504 May 7 00:17 . drwxr-xr-x@ 3 j3ssie staff 96 May 7 00:17 .. -rw-r--r
--@ 1 j3ssie staff 141 May 7 00:17 .bazelignore -rw-r--r--@ 1 j3ssie staff 11047 May 7 00:17 .bazelrc -rw-r--r--@ 1
j3ssie staff 6 May 7 00:17 .bazelversion -rw-r--r--@ 1 j3ssie staff 30 May 7 00:17 .codespellignore -rw-r--r--@ 1
j3ssie staff 321 May 7 00:17 .codespellrc drwxr-xr-x@ 4 j3s...
No existing audit state. This is the OpenAI Codex repo. Setting up the audit environment.
→ Bash ("command":"git checkout audit 2\u003e/dev/null || git checkout -b audit","description":"Create or checkou
```

# Agentic scan with autopilot

```
vigolium agent autopilot -t http://localhost:3000 --source ~/Desktop/vuln-apps/juice-shop --verbose --archon=off --no-prescan
~/Desktop/demo 11m13s
j3ssie ► vigolium agent autopilot -t http://localhost:3000 --source ~/Desktop/vuln-apps/juice-shop --verbose --archon=off -
-no-prescan
◆ olium autopilot — provider=codex-oauth model=gpt-5.5
◆ Session: ~/.vigolium/agent-sessions/344f42be-0735-42dc-ae44-47b9069bcf53
Plan:
- Inspect app structure and security-relevant routes/middleware (auth, basket/order, file/download, redirects).
- Run a targeted dynamic scan against `localhost:3000` for broad coverage, then verify a few high-confidence issues manuall
y against source and live HTTP.
- Report only confirmed distinct vulnerabilities with source locations and request/response evidence, then halt.[turn done
in=4251 out=215 cached=0]

[tool ls start] path=/Users/j3ssie/Desktop/vuln-apps/juice-shop

[tool web_fetch start] url=http://localhost:3000 max_bytes=2000

[tool list_findings start] limit=20
[tool ls ok: 1694 bytes]
[tool web_fetch ok: 2457 bytes]
[tool list_findings ok: 47 bytes]
[turn done in=5786 out=33 cached=0]

[tool load_skill start] name=audit-auth
[tool load_skill ok: 3205 bytes]
[turn done in=6661 out=158 cached=0]

[tool grep start] pattern=login|authentication|authorize|jwt|token|password|redirect|basket|file|ftp glob=*.ts ignore_case=
true max_matches=200 path=/Users/j3ssie/Desktop/vuln-apps/juice-shop
[tool grep ok: 626631 bytes]

[tool glob start] root=/Users/j3ssie/Desktop/vuln-apps/juice-shop/routes pattern=**/*.ts
[tool glob ok: 3976 bytes]
```

# Autopilot attack planning

```
~/Desktop/demo
[tool report_finding ok: 114 bytes]
[turn done in=10847 out=63 cached=27136]

[tool halt_scan start] reason=source and live target audited across auth, SQL injection, IDOR, redirect, fi...
[tool halt_scan ok: 66 bytes]
Audit complete.

Confirmed and reported 8 vulnerabilities:

- Critical: Login SQL injection bypasses authentication and returns admin JWT
- High: Product search UNION SQL injection exposes user password hashes
- High: Password change succeeds without verifying current password
- High: Profile image URL upload performs authenticated SSRF to internal URLs
- Medium: Basket endpoint allows authenticated users to read other users' baskets
- Medium: Redirect allowlist bypass via substring match
- Medium: FTP file server null-byte suffix bypass exposes backup files
- Medium: Unauthenticated support log download exposes access logs

Also ran a balanced scanner pass against `http://localhost:3000`; it produced 5 lower-severity/header/fingerprint findings
that were persisted by the scanner and not re-reported manually. Scope covered across source and live validation; halted.[t
urn done in=3256 out=168 cached=34816]

◆ autopilot complete
  findings: 8
  duration: 2m1s
  halt:     source and live target audited across auth, SQL injection, IDOR, redirect, file/log disclosure, and SSRF; 8 co
nfirm
ed findings reported plus scanner findings reviewed; no further high-value attack surface to probe in this pass
  session:  ~/ .vigolium/agent-sessions/344f42be-0735-42dc-ae44-47b9069bcf53

~/Desktop/demo 2m1s
j3ssie ▶
```

# Autopilot finding validation

```
~/Desktop/demo
j3ssie ► vigolium findings
◆ Showing 1-13 of 13 Findings
+ Severity: Critical:1 High:3 Medium:4 Low:2 Suspect:0 Info:3
+ Confidence: Certain:11 Firm:1 Tentative:1
```

Id	Sev...	Module	Short_desc	Type	Source	Url / repo name	Matched_at
13	high	olium autopilot	Profile image URL upload performs aut...	ai-age	autopilot	http://localhost:3000/profile/image/url	
12	high	olium autopilot	Password change succeeds without veri...	ai-age	autopilot	http://localhost:3000/rest/user/change-passw	
11	mediu	olium autopilot	Unauthenticated support log download ...	ai-age	autopilot	http://localhost:3000/support/Logs/access.Lo	
10	mediu	olium autopilot	FTP file server null-byte suffix bypa...	ai-age	autopilot	http://localhost:3000/ftp/package.json.bak%2	
9	mediu	olium autopilot	Redirect allowlist bypass via substri...	ai-age	autopilot	http://localhost:3000/redirect?to=https%3A%2	
8	mediu	olium autopilot	Basket endpoint allows authenticated ...	ai-age	autopilot	http://localhost:3000/rest/basket/2	
7	high	olium autopilot	Product search UNION SQL injection ex...	ai-age	autopilot	http://localhost:3000/rest/products/search	
6	criti	olium autopilot	Login SQL injection bypasses authenti...	ai-age	autopilot	http://localhost:3000/rest/user/login	
5	low	CORS Headers Detect	Passively detects permissive CORS hea...	passiv	dynamic-asse	http://localhost:3000/	http://localhost:3000/
4	info	Express.js Likely (Weak ETag)	Identifies Express.js and NestJS appl...	passiv	dynamic-asse	http://localhost:3000/	http://localhost:3000/
3	info	Permissions Policy Detect	Detects missing or overly permissive ...	passiv	dynamic-asse	http://localhost:3000/	http://localhost:3000/
2	low	Referrer Policy Detect	Detects missing or weak Referrer-Poli...	passiv	dynamic-asse	http://localhost:3000/	http://localhost:3000/
1	info	Security Headers Missing	Detects missing HTTP security headers...	passiv	dynamic-asse	http://localhost:3000/	http://localhost:3000/

```
~/Desktop/demo
j3ssie ►
```

# Agentic scan with swarm

```
~/.Desktop/demo
j3ssie ▶ vigolium scan -t http://localhost:3000 --stateless --format jsonl,html -o demo
└─ vigolium v0.1.0-alpha - Crafted with <3 by @j3ssie

  └─ Tip: run vigolium traffic list and vigolium findings list to view ingested data and vulnerabilities
  └─ Tip: run each phase separately via vigolium run <phase> (e.g. vigolium run dynamic-assessment)

+ Native Scan Configuration
  └─ Stateless mode: using temporary database
  └─ Project: 00000000-0000-0000-0000-000000000001
  └─ Strategy: balanced
  └─ Targets: 1 (CLI: http://localhost:3000)
  └─ Phases: * ExternalHarvest | ✓ Spidering (6m0s, x0.1) | ✓ Discovery (30m0s, x0.5)
            ✓ KnownIssueScan (1h0m0s, x1.0) | ✓ DynamicAssessment (1h0m0s, x1.0) | * SAST
  └─ Heuristics: basic probe target root pages to detect content type (HTML, JSON, blank) and skip spidering for non-HTML targets
  └─ Speed: concurrency=50 | rate-limit=100 | max-per-host=20
  └─ Scope: origin=relaxed | ignore-static=true host must contain the target's keyword (e.g. "example")
  └─ Modules: 143 active, 91 passive

+ Scan started 70fa35cd-9657-4b7b-9cca-b4c7cfb43d92 target: http://localhost:3000

+ HeuristicsCheck probing CLI target root pages to optimize phase selection
  └─ Level: basic | Targets: 1
  └─ Elapsed: 0s

+ Spidering browser-based crawling to discover dynamic content and API endpoints
  └─ Config: max-duration=6m0s, strategy=adaptive, headless=true (duration_factor=0.1)
  └─ Targets: 1 (1 CLI | 0 HTTP Records)
) spider [200] GET text/html http://localhost:3000/
) spider [200] GET application/javascript http://localhost:3000/chunk-24EzLz4I.js
) spider [200] GET application/javascript http://localhost:3000/chunk-T3PSKZ45.js
) spider [200] GET application/javascript http://localhost:3000/chunk-TWZWSB45.js
) spider [200] GET application/javascript http://localhost:3000/scripts.js
) spider [200] GET application/javascript http://localhost:3000/polyfills.js
) spider [200] GET application/javascript http://localhost:3000/chunk-LHKS7QUN.js
) spider [200] GET application/javascript http://localhost:3000/chunk-4MYPPGW.js
) spider [200] GET application/javascript http://localhost:3000/main.js
) spider [200] GET application/json http://localhost:3000/assets/i18n/en.json
```

# Burp integration

---

# Native scan

```
~/.Desktop/demo
j3ssie ▶ vigolium scan -t http://localhost:3000 --verbose --stateless --format jsonl,html --skip spidering -o sample
2026-04-23T11:43:02.455+0800 INFO cli/scan.go:295 Phases skipped {"skip": ["spidering"]}
4 vigolium v0.1.0-alpha - Crafted with <3 by @j3ssie

4 Tip: run vigolium traffic list and vigolium findings list to view ingested data and vulnerabilities
4 Tip: run each phase separately via vigolium run <phase> (e.g. vigolium run dynamic-assessment)

+ Native Scan Configuration
  ♦ Stateless mode: using temporary database (/var/folders/2k/z4j3lfxj5fv7nr20hswc8sj8r0000gn/T/vigolium-stateless-3143673612.sqlite)
  ♦ Project: 00000000-0000-0000-0000-000000000001
  ♦ Strategy: balanced
  ♦ Targets: 1 (CLI: http://localhost:3000)
  ♦ Phases: * ExternalHarvest | * Spidering | ✓ Discovery (30m0s, x0.5)
    ✓ KnownIssueScan (1h0m0s, x1.0) | ✓ DynamicAssessment (1h0m0s, x1.0) | * SAST
  ♦ Heuristics: basic probe target root pages to detect content type (HTML, JSON, blank) and skip spidering for non-HTML targets
  ♦ Speed: concurrency=50 | rate-limit=100 | max-per-host=20
  ♦ Scope: origin=relaxed | ignore-static=true host must contain the target's keyword (e.g. "example")
  ♦ Modules: 143 active, 91 passive

4 Tip: view scope details via vigolium config ls scope
4 Tip: view scanning pace via vigolium config ls scanning_pace
4 Tip: enrich KnownIssueScan targets with discovered paths via vigolium config known_issue_scan.enrich_targets=true

+ Scan started 95605bf8-f3c1-4d4e-948c-ea9ca67b14f4 target: http://localhost:3000

+ HeuristicsCheck probing CLI target root pages to optimize phase selection
  ♦ Level: basic | Targets: 1
2026-04-23T11:43:02.569+0800 INFO runner/heuristics.go:60 HeuristicsCheck: target passed {"target": "http://localhost:3000", "content_type": "html"}
  ♦ HeuristicsCheck: target passed | Target: http://localhost:3000 | Content-Type: html
  ♦ Elapsed: 0s

+ Discovery ingest inputs and discover directories, files, and hidden endpoints via Deparos content discovery
  ♦ Sources: deparos-discover=true
  ♦ Speed: concurrency=50, max-per-host=20, max-duration=30m0s (duration_factor=0.5)
  ♦ Targets: 1 (1 CLI | 0 HTTP Records)
    ) http://localhost:3000
```

# Vigolium CLI findings

```
.../Desktop/external/Vigolium
vigolium main*
j3ssie ► vigolium findings
◆ Showing 1-62 of 62 Findings
+ Severity: Critical:5 High:11 Medium:20 Low:4 Suspect:2 Info:20
+ Confidence: Certain:6 Firm:43 Tentative:13
```

Id	Se...	Module	Short_desc	Type	Source	Url / repo name	Matched_at
87	crit	Code Execution (RCE)	Detects OS command injection via time...	active	audit	http://localhost:3000/rest/products/search	http://localhost:3000/rest/prod
85	crit	Code Execution (RCE)	Detects OS command injection via time...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
83	crit	Code Execution (RCE)	Detects OS command injection via time...	active	audit	http://localhost:3000/api/Quantities/	http://localhost:3000/api/Quant
82	medi	SSTI Detection	Diff-based SSTI detection via error r...	active	audit	http://localhost:3000/rest/products/search	http://localhost:3000/rest/prod
81	medi	Smart Behavior Detection	Diff-based injection detection via be...	active	audit	http://localhost:3000/rest/products/search	http://localhost:3000/rest/prod
80	medi	Nginx Path Escape Detection	Diff-based Nginx path escape detectio...	active	audit	http://localhost:3000/rest/continue-code	http://localhost:3000/rest/cont
79	medi	Nginx Path Escape Detection	Diff-based Nginx path escape detectio...	active	audit	http://localhost:3000/ftp/	http://localhost:3000/ftp/
77	susp	Input Behavior Probe	Detects behavior changes from header,...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
75	susp	Input Behavior Probe	Detects behavior changes from header,...	active	audit	http://localhost:3000/rest/products/search	http://localhost:3000/rest/prod
74	high	NoSQL Operator Injection	Detects MongoDB operator injection (\$...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
73	high	NoSQL Operator Injection	Detects MongoDB operator injection (\$...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
72	high	NoSQL Operator Injection	Detects MongoDB operator injection (\$...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
71	high	NoSQL Boolean-based Injection	Detects MongoDB operator injection (\$...	active	audit	http://localhost:3000/socket.io/?EIO=4&tr	http://localhost:3000/socket.io
70	high	NoSQL Operator Injection	Detects MongoDB operator injection (\$...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
69	high	NoSQL Operator Injection	Detects MongoDB operator injection (\$...	active	audit	http://localhost:3000/api/Feedbacks/	http://localhost:3000/api/Feedb
68	low	Info Disclosure Detect	Detects information disclosure patter...	passiv	audit	http://localhost:3000/ftp	http://localhost:3000/ftp
67	low	Directory Listing Detected ...	Passively detects directory listing e...	passiv	audit	http://localhost:3000/ftp	http://localhost:3000/ftp
66	high	NoSQL	Detects MongoDB operator	active	audit	http://localhost:3000/socket.io/?EIO=4&tr	http://localhost:3000/socket.io

# Vigolium CLI traffic

```
.../Desktop/external/Vigolium
vigolium main*
j3ssie ► vigolium traffic 'text/html'
Showing 1-41 of 41 records
```

Host	Method	Path	Status	Content_type	Size	Words	Title	Source
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		finding
http://localhost:3000	GET	./ftp/juice-shop	200	text/html; charset=UTF-8	0	45		finding
http://localhost:3000	POST	/socket.io/?EIO=4&transport=polling&...	200	text/html	2	20		finding
http://localhost:3000	POST	/socket.io/?EIO=3&C%2510000000.times%...	200	text/html	2	20		finding
http://localhost:3000	POST	/socket.io/?EIO=4&transport=polling&...	200	text/html	2	20		finding
http://localhost:3000	POST	/socket.io/?EIO=4&transport=33C%25100...	200	text/html	2	20		finding
http://localhost:3000	POST	./api/Feedbacks/	200	text/html; charset=UTF-8	0	45		finding
http://localhost:3000	POST	./api/Feedbacks/	200	text/html; charset=UTF-8	0	45		finding
http://localhost:3000	PUT	HTTP/1.1	200	text/html; charset=UTF-8	0	45		finding
http://localhost:3000	POST	HTTP/1.1	200	text/html; charset=UTF-8	0	45		finding
http://localhost:3000	GET	/chunk-BBQJZN7H.js	200	application/javascript...	236053	5862		deparos
http://localhost:3000	GET	/scripts	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/styles	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/favicon_js/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/assets/public	301	text/html; charset=UTF-8	163	49		Redirecting
http://localhost:3000	GET	/main	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/ftp	200	text/html; charset=utf-8	11296	969		listing directory /ftp
http://localhost:3000	GET	/public	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/favicon_js/xhtmll	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/ftp/juice-shop	403	text/html; charset=utf-8	1935	188		Error: Only .md and .pdf fi...
http://localhost:3000	GET	/ftp/	200	text/html; charset=utf-8	11253	971		listing directory /ftp/
http://localhost:3000	GET	/assets/favicon_js/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/public/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/assets/scripts/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/scripts/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/assets/styles/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/assets/polyfills/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/favicon_js	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	/styles/	200	text/html; charset=UTF-8	75002	3650		deparos
http://localhost:3000	GET	HTTP/1.1	200	text/html; charset=UTF-8	0	45		scanner
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		spidering
http://localhost:3000	POST	/api/Feedbacks/	401	text/html; charset=utf-8	42	38		spidering
http://localhost:3000	GET	/chunk-BBQJZN7H.js	304	application/javascript...	236053	5857		spidering
http://localhost:3000	POST	/rest/user/login	401	text/html; charset=utf-8	26	35		spidering
http://localhost:3000	GET	/chunk-BBQJZN7H.js	200	application/javascript...	236053	5866		spidering

# Thank you

---

Thanks for listening. Connect with me on LinkedIn for updates, questions, and follow-up.



[linkedin.com/in/jessie-aiho](https://www.linkedin.com/in/jessie-aiho) / [jessie@vigolium.com](mailto:jessie@vigolium.com)