ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 20 / 30 · AI OFFENSE

Specialized AI Vulnerability Agents
From Patch Diff to Container Swarm

How DeepAudit, AutoCVE, Cairn, AutoHunter, Strix, RedAmon and XingRin split vulnerability work across 285 endpoints

Two-week observation window ending 18 November 2026 · Prepared November 2026
Full Report · Public edition · Version 1.1

285
Endpoints running one of seven specialized AI vulnerability agents, on 250 hosts

17
Confirmed controller endpoints to block, on 13 dedicated cloud addresses (6.0% of endpoints)

118
AutoCVE endpoints: 7 confirmed on port 8000, 2 on dynamic ISPs, 109 candidates

61
DeepAudit endpoints, all on review: 37 confirmed dual-use, 2 on dynamic ISPs, 22 candidates

What Is in This Report. 11 chapters and two appendices.

How DeepAudit, AutoCVE, Cairn, AutoHunter, Strix, RedAmon and XingRin split vulnerability work across 285 endpoints

01	Executive Summary	04
02	How the Specialized Agents Fit Together	07
03	Source Mining and Advisory Synthesis: DeepAudit and AutoCVE	10
04	Attack-Surface Discovery: Cairn and AutoHunter	14
05	Container Validation Swarms and Co-Hosting	17
06	Separating Hostile Controllers from Benign Lookalikes	20
07	What Defenders Should Do	22
08	Detection Rules	24
09	Indicators	27
10	Scope and Limits	30
11	Sources	32
A	Indicators of compromise (17)	34
B	Detection rules (4 files)	35

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense found 285 endpoints on 250 hosts running seven task-specialized AI vulnerability agents. Each tool does one job: source-code auditing, turning advisories into checks, mapping attack surface, or validating targets from containers. Only 17 endpoints on 13 dedicated cloud hosts are confirmed controllers. DeepAudit is dual-use and stays on review. Block the 13 confirmed addresses, send DeepAudit hits to policy review, and remove exposed `.git` folders from your perimeter.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense found 285 endpoints on 250 hosts running seven task-specialized AI vulnerability agents. Each tool does one job: source-code auditing, turning advisories into checks, mapping attack surface, or validating targets from containers. Only 17 endpoints on 13 dedicated cloud hosts are confirmed controllers. DeepAudit is dual-use and stays on review. Block the 13 confirmed addresses, send DeepAudit hits to policy review, and remove exposed `.git` folders from your perimeter.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 04

01 Executive Summary

Earlier AI offense consoles wrapped a chatbot around many tools at once. The agents in this report are narrower. Each one automates a single slow step in vulnerability work. This report maps where these agents run, how they are built, and which signals let defenders find them without blocking legitimate security teams.

1. **AutoCVE turns new advisories into HTTP checks on 118 endpoints.**
   AutoCVE is the largest family here, at 41.4% of all endpoints. Seven dedicated cloud controllers on port 8000 are confirmed. Two more confirmed instances sit on dynamic telecom lines, and 109 uncorroborated candidates, 85 of them behind one shared CDN provider, are not blocked.

2. **DeepAudit audits source code but is dual-use, so nothing is blocked.**
   DeepAudit runs on 61 endpoints across 49 hosts, pairing web consoles with Python analysis backends. Of these, 39 are confirmed DeepAudit services: 37 on cloud hosts and 2 on dynamic ISPs. None goes on a block list, because enterprise AppSec teams use the same open-source tool.

3. **Recon miners and container swarms extend the chain to live targets.**
   Cairn (17 endpoints) and AutoHunter (21) map routes and parameters. Strix (40), RedAmon (17) and XingRin (11) run web and API checks from containers. Three hosts carry two agent families side by side.

Three Decisions for Leaders

1. **Block confirmed controllers, review dual-use tools.** Block the 13 confirmed controller addresses. Send DeepAudit, CDN-fronted candidates and dynamic telecom addresses to analyst watchlists instead.
2. **Remove source leaks and shorten N-day patch windows.** Find and remove exposed `.git` metadata and source archives, which feed source-code miners. Patch internet-facing gateways listed in public advisory feeds faster.
3. **Pair scan bursts with agent API and process signals.** Alert when unapproved Python, Node or container processes connect out on ports 3000, 8000, 8083, 8084 to 8093 and 18800.

The Study in Numbers

Number	What it is
285	Endpoints running one of seven specialized AI vulnerability agents, on 250 hosts
17	Confirmed controller endpoints to block, on 13 dedicated cloud addresses (6.0% of endpoints)
118	AutoCVE endpoints: 7 confirmed on port 8000, 2 on dynamic ISPs, 109 candidates
61	DeepAudit endpoints, all on review: 37 confirmed dual-use, 2 on dynamic ISPs, 22 candidates
38	Recon endpoints: Cairn (17, port 8000) and AutoHunter (21, mostly port 18800)
1.74x	Strix endpoints per host (40 on 23 hosts), the highest multi-port density in the study
3	Hosts that run two specialized AI agent families on separate ports

How to use this report.

Security leaders can read Chapter 1, Chapter 6 and Chapter 7. AppSec and cloud teams should read Chapters 3 to 5 on source mining, recon and swarms. Detection engineers and threat hunters should start with Chapters 7 to 9.

02

CHAPTER 02

How the Specialized Agents Fit Together

IN SHORT

Second-generation tools split vulnerability work into narrow stages instead of one chat console. We describe four stages, which is an analytical model built from the tools' documented APIs.

INSIDE THIS CHAPTER

From one console to narrow tools 07

Counts by family 08

Where the agents run 08

02 How the Specialized Agents Fit Together

From one console to narrow tools

First-generation AI offense frameworks wrapped a large language model (LLM) inside one chat dashboard. Those dashboards hit context limits and produced noisy command loops. The tools in this report each do one narrow job instead.

Stage	Families	What the stage does
1. Source auditing	DeepAudit	Traces untrusted input through source code to dangerous sinks
2. Advisory to check	AutoCVE	Turns a public advisory or patch diff into an HTTP verification template
3. Surface discovery	Cairn, AutoHunter	Maps subdomains, API routes and query parameters
4. Container validation	Strix, RedAmon, XingRin	Runs web and API checks from isolated containers

This four-stage chain is a model, not an observed campaign. Each family was observed on its own. Only three hosts link two families, and no single operator was seen running the whole chain. We saw controllers, not activity on any target system.

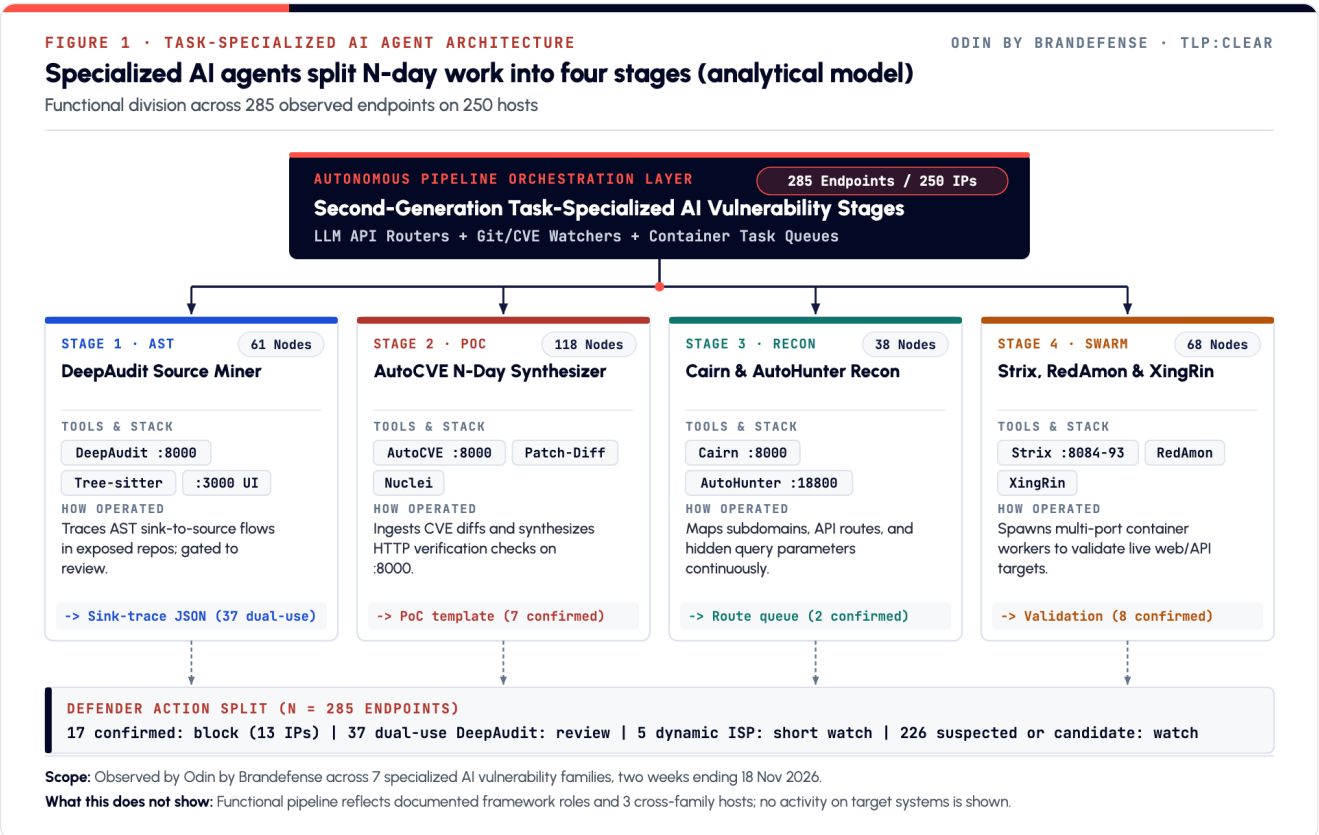


Figure 1: Specialized AI agents split N-day work into four stages (analytical model)

Counts by family

Each endpoint is an IP, port and family combination, labelled confirmed, suspected or candidate.

Family	Stage	Endpoints	Hosts	Per host	Block	Watch or review
AutoCVE	Advisory to check	118	116	1.02x	7	111 (2 dynamic ISP, 109 candidates)
DeepAudit	Source auditing	61	49	1.24x	0	61 (37 dual-use, 2 dynamic ISP, 22 candidates)
Strix	Container validation	40	23	1.74x	5	35 candidates
AutoHunter	Surface discovery	21	21	1.00x	1	20 candidates
Cairn	Surface discovery	17	17	1.00x	1	16 (1 suspected, 15 candidates)
RedAmon	Container validation	17	17	1.00x	3	14 candidates
XingRin	Container validation	11	10	1.10x	0	11 (1 dynamic ISP, 3 suspected, 7 candidates)
Total		**285**	**250**		**17**	**268**

The per-family host counts add up to 253. Three hosts run two families, so the study covers 250 distinct hosts.

Where the agents run

The United States hosts 149 endpoints (52.3%), China 59 (20.7%) and Hong Kong 24 (8.4%). Singapore (9), Germany (4), Canada (4), Finland (4) and the Netherlands (3) follow.

The US total needs care. Of the 100 US AutoCVE records, 85 sit behind one shared proxy provider, Uscloud Inc, on port 3000 with a `CN=Cloudflare` certificate. Confirmed controllers are spread more widely.

Hosting network	Confirmed endpoints
Shenzhen Tencent Computer Systems	5
IXROOT LTD	5 (all on one host)
Alibaba Cloud	2
HostPapa	1
cognetcloud INC	1
Oracle Cloud	1
Contabo	1
Leaseweb Singapore	1

CHAPTER 03

Source Mining and Advisory Synthesis: DeepAudit and AutoCVE

IN SHORT

DeepAudit and AutoCVE automate the two slowest manual steps in exploit development. Together they account for 179 of the 285 endpoints (62.8%).

INSIDE THIS CHAPTER

DeepAudit: multi-agent source-code auditing 10
Why DeepAudit is not blocked 11
AutoCVE: from advisory to HTTP check 11
The CDN collision on port 3000 12

03 Source Mining and Advisory Synthesis: DeepAudit and AutoCVE

For AppSec and perimeter teams, these two tools shrink the time between a code leak or a patch release and the first automated probe. Figure 2 sets out each family's ports, API routes, target surfaces and matching detection rules.

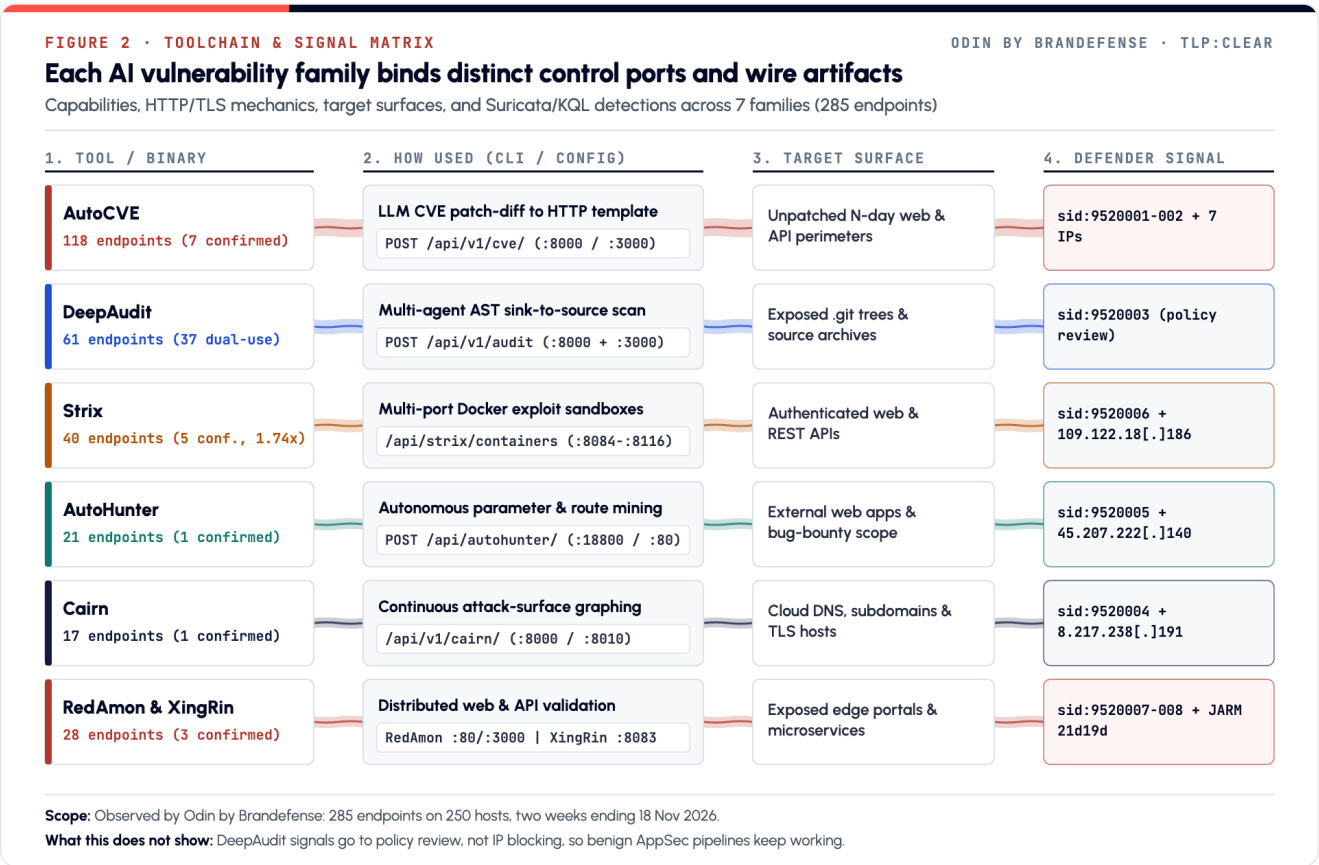


Figure 2: Each AI vulnerability family binds distinct control ports and wire artifacts

DeepAudit: multi-agent source-code auditing

DeepAudit runs several LLM sub-agents over Tree-sitter abstract syntax trees (AST). An AST is a structured map of source code. The agents trace user input from HTTP controllers into database queries, URL fetchers, deserializers and authentication code.

Engineering teams use DeepAudit for pre-release code review. An attacker can use the same tool on exposed `.git` folders, leaked source archives or open-source dependencies. Code submitted to `/api/v1/audit` comes back as candidate SQL injection, server-side request forgery and authentication bypass paths.

Port	DeepAudit endpoints	Role
8000	35	FastAPI analysis backend
3000	15	Node or React web console
8001	2	Backend
8888	2	Backend
Other single ports, such as 3500	1 each	Console or backend

Of the 49 DeepAudit hosts, 12 (24.5%) expose a web console and a backend on the same server. Examples sit on HostPapa (US), Hangzhou Alibaba (CN) and Netsec Limited (HK); their addresses are withheld. DeepAudit endpoints are in China (23), the United States (18) and Hong Kong (10).

Why DeepAudit is not blocked

From the outside, an approved DevSecOps code review server looks the same as an attacker mining a leaked repository. None of the 49 DeepAudit hosts also runs a confirmed offensive controller. So all 37 confirmed cloud instances stay on a dual-use watchlist, not a block list.

Two more confirmed instances sit on China Telecom and go to a 24-hour watchlist. Defenders should alert on behavior instead: an unapproved internal host posting corporate code or a `.git` archive to an external `/api/v1/audit` endpoint.

AutoCVE: from advisory to HTTP check

AutoCVE watches structured vulnerability feeds, such as the NVD REST API and vendor git commits. It turns a patch diff into a proof-of-concept HTTP request and a `nuclei`-compatible verification template. `nuclei` is a common template-based vulnerability scanner.

Port	AutoCVE endpoints
3000	105
8000	9
13000	2
80	1
8088	1

All nine confirmed AutoCVE backends listen on TCP 8000. Seven sit on dedicated cloud hosting at Shenzhen Tencent, Hangzhou Alibaba and HostPapa, and are block-ready. The other two sit on China Telecom and China Mobile and go to a 24-hour watchlist. Both also expose a 3000 console.

Endpoint	Country	Hosting	Seen
150.158.23[.]244:8000	CN	Shenzhen Tencent Computer Systems	13 Nov 2026
49.232.241[.]96:8000	CN	Shenzhen Tencent Computer Systems	13 Nov 2026
107.175.69[.]167:8000	US	HostPapa	13 Nov 2026
150.158.93[.]134:8000	CN	Shenzhen Tencent Computer Systems	13 Nov 2026
43.130.10[.]7:8000	US	Shenzhen Tencent Computer Systems	13 Nov 2026
43.159.59[.]133:8000	SG	Shenzhen Tencent Computer Systems	13 Nov 2026
8.147.132[.]181:8000	CN	Hangzhou Alibaba Advertising	13 Nov 2026

The CDN collision on port 3000

Most AutoCVE records are not confirmed. Of the 109 candidates, 85 listen on port 3000 inside Uscloud Inc in the United States. They present `CN=Cloudflare` certificates and share one JARM value. JARM is a fingerprint of how a server answers TLS handshakes.

A generic Node or React page on port 3000 behind a shared CDN edge says little about who runs it. Blocking those addresses would break unrelated web tenants. These candidates stay on watch until an AutoCVE API route is seen on the same endpoint.

04

CHAPTER 04

Attack-Surface Discovery: Cairn and AutoHunter

IN SHORT

A synthesized check needs a matching live target. Cairn and AutoHunter keep mapping subdomains, API routes and query parameters so that checks have somewhere to land.

INSIDE THIS CHAPTER

Cairn attack-surface controllers 14

AutoHunter parameter mining on port 18800 15

04 Attack-Surface Discovery: Cairn and AutoHunter

Figure 3 shows the modeled sequence from advisory and source input, through recon, to container validation. It also marks where each detection rule sits.

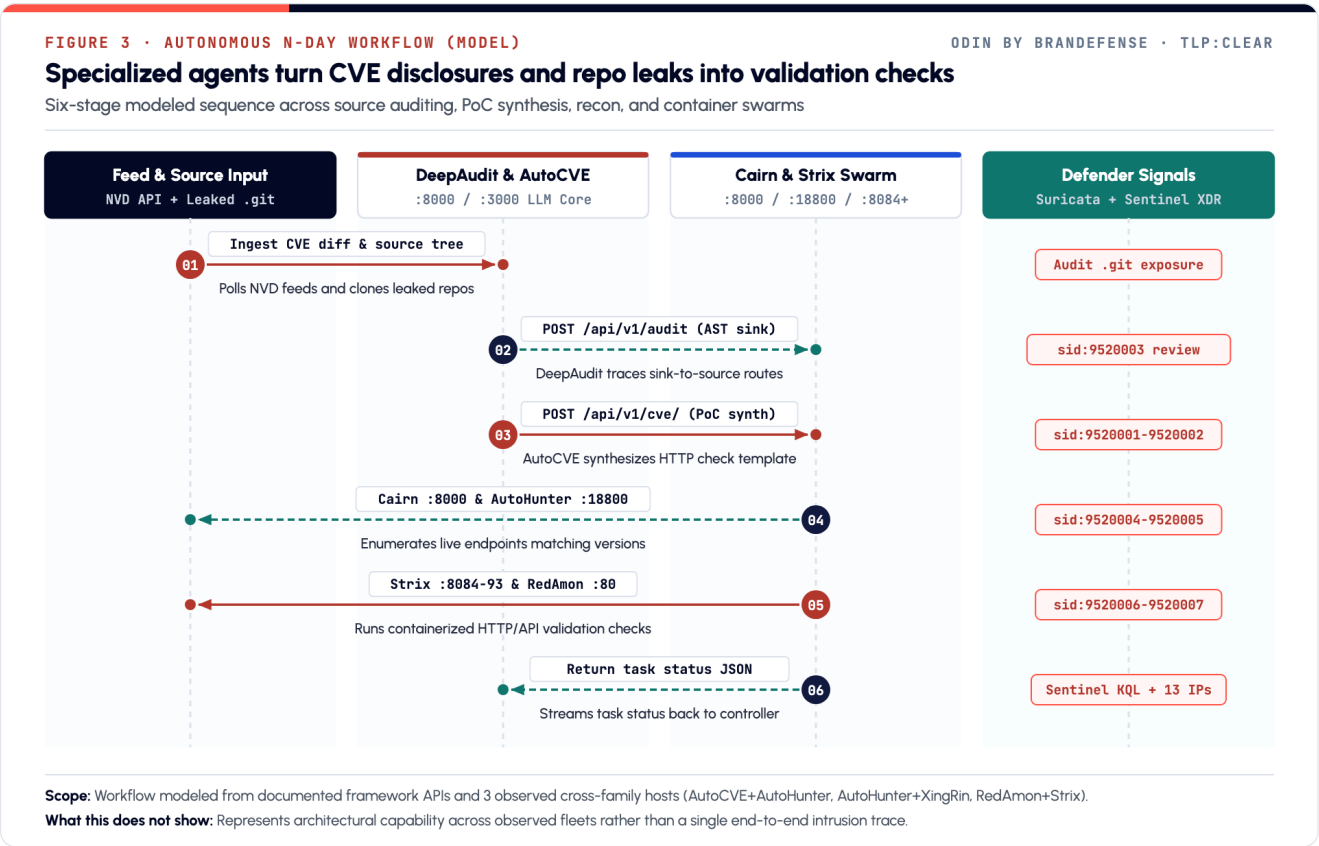


Figure 3: Specialized agents turn CVE disclosures and repo leaks into validation checks

Cairn attack-surface controllers

Cairn keeps an asset graph built from DNS zones, TLS certificates and HTTP banners. It exposes worker coordination APIs on ports 8000 and 8010. Cairn runs 17 distinct endpoints on port 8000.

Cairn endpoints by country	Count
Netherlands	3
Hong Kong	3
Canada	3
Finland	2
United States	2

The main hosting networks are The Constant Company (4), Google LLC (3) and OVH SAS (2). One Cairn controller is confirmed: 8.217.238[.]191:8000 on Alibaba (US) Technology in Hong Kong, confirmed on 18 November 2026.

One more US instance, on Advin Services LLC, is suspected. The other 15 are uncorroborated candidates on port 8000. Their addresses are withheld.

AutoHunter parameter mining on port 18800

AutoHunter automates headless bug-bounty recon. It extracts endpoints from JavaScript and queues parameters for fuzzing. It runs on 21 endpoints, each on a separate host.

Fourteen of the 21 (66.7%) use the default control port 18800. Single listeners sit on 80, 443, 8080, 18081, 23253, 50000 and 60707. China hosts 14 endpoints, 11 of them on Shenzhen Tencent. Hong Kong hosts 6, led by 4 on cognetcloud INC.

One AutoHunter controller is confirmed: 45.207.222[.]140:80 on cognetcloud INC in Hong Kong, seen on 13 November 2026. The other 20 are uncorroborated candidates on 18800 and high ports, and stay on watch.

05

CHAPTER 05

Container Validation Swarms and Co-Hosting

IN SHORT

Strix, RedAmon and XingRin run web and API checks inside short-lived containers. Operators put several container listeners on one host, and sometimes put two agent families on one server.

INSIDE THIS CHAPTER

Strix multi-port container hubs 17

RedAmon web and API validators 18

XingRin scanner nodes 18

Three hosts run two families 18

05 Container Validation Swarms and Co-Hosting

Container validation agents start an isolated sandbox for each target job. Each sandbox binds its own high port and streams HTTP replay logs and session state back to a parent controller. Figure 4 shows the observed port bindings and runtimes on multi-port and cross-family hosts.

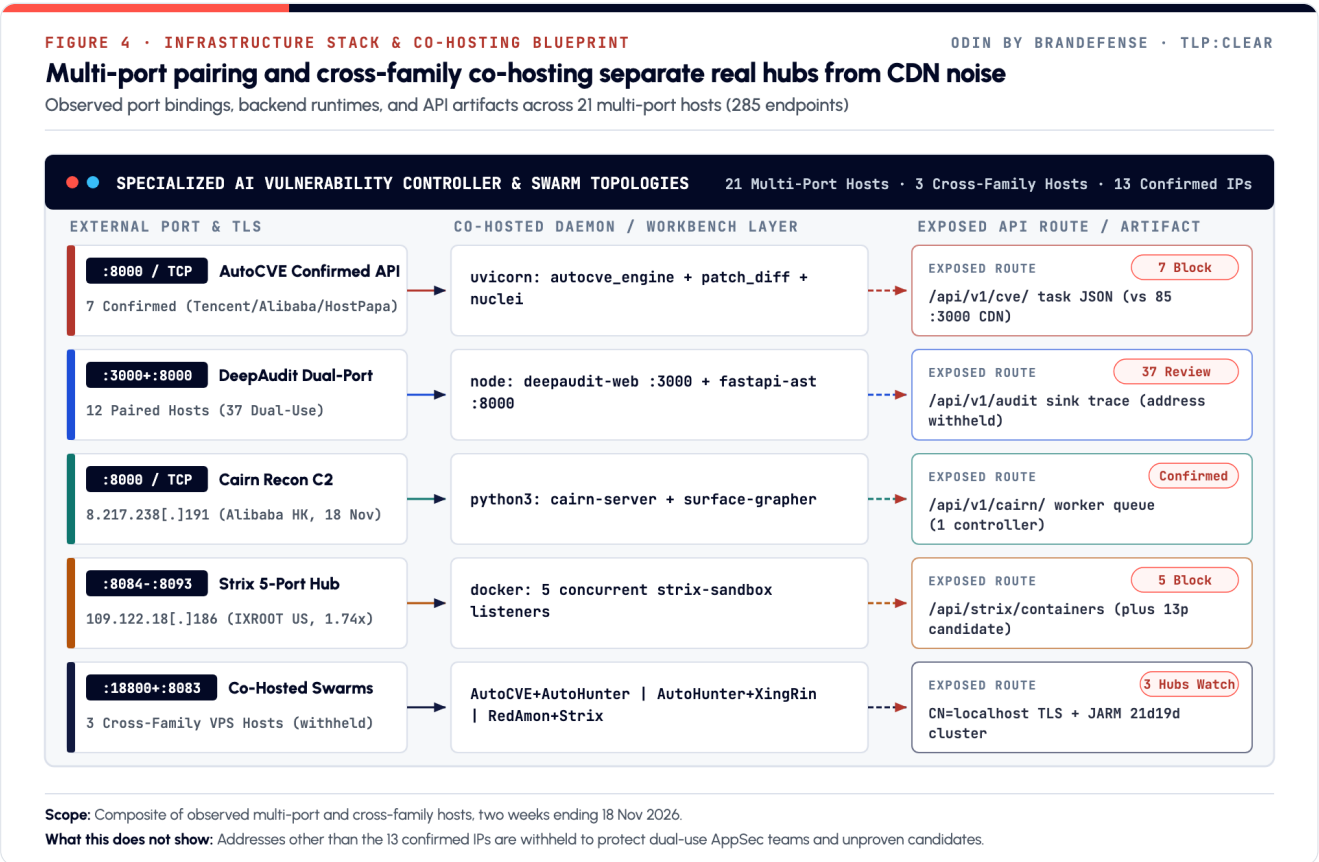


Figure 4: Multi-port pairing and cross-family co-hosting separate real hubs from CDN noise

Strix multi-port container hubs

Strix has the highest port-to-host density in the study: 40 endpoints on 23 hosts (1.74x). IXROOT LTD in the United States hosts 18 of the 40 (45.0%), and Hetzner Online GmbH hosts 4. Two IXROOT hubs show how Strix gives each task its own Docker sandbox.

Hub	Ports	Status
109.122.18[.]186 (IXROOT LTD, US)	8084, 8085, 8086, 8089, 8093	Five confirmed Strix listeners, seen 13 November 2026; block
Address withheld (IXROOT LTD, US)	13 ports: 8081, 8088, 8092, 8096, 8101, 8102, 8108 to 8112, 8114, 8116	Candidates seen 16 November 2026; watch until an HTTP route confirms them

RedAmon web and API validators

RedAmon has 17 endpoints, each on a separate host. Ten are candidate operator dashboards on port 3000. Agent listeners sit on 80 and 443. Three controllers on port 80 are confirmed, in three regions.

Endpoint	Country	Hosting	Seen
147.224.215[.]112:80	US	Oracle Corporation	13 Nov 2026
169.58.136[.]189:80	DE	Contabo GmbH	13 Nov 2026
23.111.14[.]152:80	SG	Leaseweb Singapore	13 Nov 2026

XingRin scanner nodes

XingRin has 11 endpoints on 10 hosts in China and Hong Kong. Seven sit on port 8083. Five of those present a self-signed `CN=localhost` certificate with the same JARM value:

```
21d19d00021d21d00042d43d00000091f9827a8676a9d9f27d421962a09b5d
```

The one confirmed XingRin instance sits on China Telecom, a dynamic telecom line, so it goes to a 24-hour watchlist. Three more XingRin endpoints are suspected. All XingRin addresses are withheld.

Three hosts run two families

Three servers run two specialized agent families at once. This is why 253 family-host pairs come down to 250 distinct hosts. None of these hosts is confirmed, so their addresses are withheld.

Host	Families and ports	Hosting (country)	What the pairing suggests
Withheld	AutoCVE (3000) and AutoHunter (60707)	Shenzhen Tencent (CN)	An advisory-to-check console next to a parameter miner
Withheld	AutoHunter (18800) and XingRin (8083)	cognetcloud INC (HK)	Route discovery next to a `CN=localhost` scanner
Withheld	RedAmon (3000) and Strix (8080)	The Constant Company (KR)	Two web and API validation consoles on one VPS

These three hosts are the only concrete links between families in this study. They show that some operators combine recon and validation tools. They do not show one operator running the full four-stage chain.

06

CHAPTER 06

Separating Hostile Controllers from Benign Lookalikes

IN SHORT

Blocking every AI security tool banner would hit CDN tenants, AppSec teams and home broadband users. Only 17 of the 285 endpoints (6.0%) are block-ready.

06 Separating Hostile Controllers from Benign Lookalikes

AI vulnerability tools have three common benign or unattributable explanations. Each one calls for a different action.

Situation	Endpoints	Why not block	What to do
Dual-use DeepAudit	37 confirmed, plus 22 candidates	Enterprise AppSec teams run the same open-source auditor	Policy review; alert on internal code uploads to external `/api/v1/audit`
Shared CDN and uncorroborated candidates	222 candidates, including 85 AutoCVE on Uscloud 3000	Generic Node or React pages behind shared edges serve many unrelated tenants	Watch until a family-specific API route appears
Dynamic telecom lines	5 confirmed: 2 AutoCVE, 2 DeepAudit, 1 XingRin	Addresses on China Telecom (4) and China Mobile (1) rotate between customers	24-hour watchlist, never a permanent block
Dedicated cloud controllers	17 confirmed: AutoCVE 7, Strix 5, RedAmon 3, AutoHunter 1, Cairn 1	Confirmed offensive controller on dedicated cloud or VPS hosting	Block; re-check after 30 days

Four endpoints are suspected rather than confirmed: one Cairn and three XingRin. They stay on watch until a second signal appears.

Cloud VPS addresses can be reassigned to new customers. Block entries for the 13 confirmed addresses should be re-checked after 30 days.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

This chapter lists the actions defenders should take, in priority order.

07 What Defenders Should Do

Within 24 hours

- [ ] Block the 13 confirmed controller addresses (17 endpoints) at perimeter firewalls and proxies. Re-check them after 30 days.
- [ ] Treat any agent match on a dynamic telecom line as a 24-hour watchlist item, not a permanent block.
- [ ] Route DeepAudit matches to policy review, not to automatic blocking.

Within 72 hours

- [ ] Scan external web properties and CDN origins for exposed `.git/config` files, source-map bundles and backup archives, and remove them.
- [ ] Shorten emergency patch deadlines for internet-facing gateways listed in the CISA Known Exploited Vulnerabilities catalog and NVD.
- [ ] Rate-limit unauthenticated API routes, and add bot challenges where recon swarms probe them.

Within 7 to 30 days

- [ ] Deploy the eight Suricata signatures, routing SID 9520003 (DeepAudit) to a policy queue.
- [ ] Enable the Sentinel KQL hunt and the three Sigma rules, with exact host names of approved AppSec runners in the allowlists.
- [ ] Add the JARM values in Chapter 9 to external attack-surface hunting.
- [ ] Watch container hosts for unapproved AutoCVE, Cairn, AutoHunter, Strix, RedAmon or XingRin worker processes.

CHAPTER 08

Detection Rules

IN SHORT

This chapter describes each published rule and where to deploy it.

INSIDE THIS CHAPTER

How the Suricata signatures work 24

How the other rules work 24

Tuning 25

08 Detection Rules

The full rules are in Appendix B. Each file targets an API route, a header or a process pattern that the agents use.

Rule file	What it catches
`odin_specialized_ai_vuln_agents.rules`	Suricata 7: eight signatures for agent API calls and check-ins across all seven families
`sentinel_autocve_cairn_and_ai_scanner_burst.kql`	Microsoft Sentinel and Defender XDR: contact with the 13 confirmed controllers, plus unapproved agent processes and high-rate egress on agent ports
`sigma_specialized_ai_exploit_agent_execution.yml`	Sigma (3 rules): unapproved AutoCVE runners, DeepAudit backends and recon or swarm worker launches on Linux
`odin_deepaudit_autocve_cairn_strix_artifacts.yar`	YARA (4 rules): configuration and orchestration files for each agent group

How the Suricata signatures work

SID	Looks for	Why
9520001	Outbound `POST /api/v1/cve/` with `"cve_id"` and `"patch_diff"` in the body, to ports 8000, 3000 or 13000	Catches an internal host sending an advisory to AutoCVE
9520002	An HTTP 200 response from ports 8000 or 3000 with `"autocve_task"` and `"template_status"`	Catches AutoCVE returning task status
9520003	Outbound `POST /api/v1/audit` with `"repo_source"` and `"sink_trace"`	Catches code sent to DeepAudit; policy alert only, since the tool is dual-use
9520004	Outbound request to `/api/v1/cairn/` with an `X-Cairn-Worker:` header, to ports 8000 or 8010	Catches Cairn worker check-ins
9520005	Outbound `POST /api/autohunter/` with `"target_scope"` in the body	Catches AutoHunter recon queue dispatch
9520006	Outbound `POST /api/strix/containers` with `"sandbox_port"` in the body	Catches Strix sandbox control
9520007	Outbound request to `/api/redamon/` with a user agent starting `RedAmon-Agent/`	Catches RedAmon agent beacons
9520008	Outbound `POST /api/xingrin/node` with an `X-XingRin-Token:` header	Catches XingRin node heartbeats

Suricata sees these signatures only in clear text. For TLS listeners on 443 or 8083, use TLS inspection or pair the signatures with the KQL hunt.

How the other rules work

The KQL query looks back 7 days. It alerts on any device that contacts one of the 13 confirmed controller addresses, whatever its name. It also flags Python, Node, `nuclei`, `httpx` or `curl` connections on agent ports from hosts outside an exact-name allowlist. Process launches that name an agent add context, and contact with 15 or more remote addresses marks a scan burst.

The query sorts each hit into one of three actions: block a confirmed controller, review an unapproved DeepAudit launch, or hunt a high-rate scanner swarm. The allowlist never hides a match on a confirmed controller address.

The first Sigma rule needs a runtime such as `python3`, `uvicorn` or `nuclei` and a command line naming `autocve`, `cve_synthesizer`, `patch_diff_agent` or `--cve-feed`. The second flags DeepAudit backends at medium severity for dual-use review. The third matches worker names such as `cairn-worker`, `strix-sandbox` or `xingrin` with port 8083. Each rule excludes only named, approved hosts.

Each YARA rule is limited to files under 512 KB and needs at least two independent strings.

YARA rule	Fires when
`ODIN_DeepAudit_MultiAgent_AST_Config`	A DeepAudit identifier, an AST marker such as `sink_to_source`, and `/api/v1/audit` appear together
`ODIN_AutoCVE_PatchDiff_Synthesizer_Manifest`	An AutoCVE identifier, a patch-diff or NVD sync marker, and `template_status` appear together
`ODIN_Cairn_AutoHunter_Recon_Queue_Config`	A Cairn header and route, or the AutoHunter route and port 18800, appear with `target_scope`
`ODIN_Strix_RedAmon_XingRin_Swarm_Compose`	The Strix route and `sandbox_port`, the RedAmon user agent and route, or the XingRin route and token header appear

Tuning

Approved internal tools are the main false positive. Add the exact host names of CI/CD static-analysis and vulnerability-management runners to the KQL and Sigma allowlists. Never use wildcard host patterns. Add Suricata `pass` rules for those runners by source address before any inline enforcement.

09

CHAPTER 09

Indicators

IN SHORT

This chapter lists the published indicators and explains what stays withheld.

INSIDE THIS CHAPTER

Confirmed endpoints to block 27

JARM fingerprints for hunting 28

09 Indicators

Appendix A holds 17 published indicators: 17 controller endpoints (IP and port) on 13 distinct addresses, and no file hashes. Two JARM fingerprints for hunting appear in this report only. All other addresses are withheld.

Type	Count	Where
Confirmed controller endpoints to block	17 (13 addresses)	Appendix A and the table below
JARM fingerprints for hunting	2	This report only
Selected watchlist, review or dual-use endpoints	9 (of 268 watch records)	Withheld; described in Chapters 3 to 6

Confirmed endpoints to block

Endpoint	Family	Country	Hosting	Seen
150.158.23[.]244:8000	AutoCVE	CN	Shenzhen Tencent	13 Nov 2026
49.232.241[.]96:8000	AutoCVE	CN	Shenzhen Tencent	13 Nov 2026
107.175.69[.]167:8000	AutoCVE	US	HostPapa	13 Nov 2026
150.158.93[.]134:8000	AutoCVE	CN	Shenzhen Tencent	13 Nov 2026
43.130.10[.]7:8000	AutoCVE	US	Shenzhen Tencent	13 Nov 2026
43.159.59[.]133:8000	AutoCVE	SG	Shenzhen Tencent	13 Nov 2026
8.147.132[.]181:8000	AutoCVE	CN	Hangzhou Alibaba	13 Nov 2026
109.122.18[.]186:8084	Strix	US	IXROOT LTD	13 Nov 2026
109.122.18[.]186:8085	Strix	US	IXROOT LTD	13 Nov 2026
109.122.18[.]186:8086	Strix	US	IXROOT LTD	13 Nov 2026
109.122.18[.]186:8089	Strix	US	IXROOT LTD	13 Nov 2026
109.122.18[.]186:8093	Strix	US	IXROOT LTD	13 Nov 2026
45.207.222[.]140:80	AutoHunter	HK	cognetcloud INC	13 Nov 2026
8.217.238[.]191:8000	Cairn	HK	Alibaba (US) Technology	18 Nov 2026
147.224.215[.]112:80	RedAmon	US	Oracle	13 Nov 2026
169.58.136[.]189:80	RedAmon	DE	Contabo	13 Nov 2026
23.111.14[.]152:80	RedAmon	SG	Leaseweb	13 Nov 2026

JARM fingerprints for hunting

JARM	Seen on	Use
`21d19d00021d21d00042d43d00000091f9827a867 6a9d9f27d421962a09b5d`	5 XingRin listeners with `CN=localhost` on 8083, and 1 DeepAudit listener	Hunt only; pair with an API route before acting
`3fd3fd0003fd3fd21c42d42d000000bdfc58c9a46 434368cf60aa440385763`	Strix (2), AutoHunter (1) and XingRin (1) listeners	Hunt only; pair with an API route before acting

Withheld: the nine selected watch endpoints. They are two DeepAudit dual-use hosts, the 13-port Strix candidate hub and three cross-family hosts. The rest are one suspected Cairn node, one suspected XingRin node and the confirmed XingRin instance on a telecom line. Every candidate and dynamic-ISP address is also withheld, as are certificate names seen only on unconfirmed hosts.

10

CHAPTER 10

Scope and Limits

IN SHORT

This chapter sets out what the evidence can show and what it cannot.

10 Scope and Limits

- **Window.** The findings cover a two-week observation window ending 18 November 2026. Some records carry earlier first-seen dates; we do not treat those as a tracking start.
- **Controllers, not intrusions.** We saw controller infrastructure. We did not see activity on any target system, and nothing here shows that a check reached or affected a target.
- **The four-stage chain is a model.** It comes from the tools' documented APIs. Only three hosts link two families, and no single operator was seen running all four stages.
- **Dual-use tools.** These agents are open-source security tools. DeepAudit stays on review because legitimate AppSec teams use it. The 17 block entries are confirmed controllers on dedicated hosting; re-check them after 30 days, since cloud addresses are reassigned.
- **Candidates are leads.** The 222 candidates are hunting leads. Do not block them without corroboration.
- **No growth claims.** The data supports counts in the window, not trends over time.
- **Rule testing.** Every published rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma and Microsoft's Kusto.Language analyzer, checked on 20 November 2026). The rules were not replayed against live traffic, so test and tune them in your own environment before production use.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. OWASP Foundation. OWASP Top 10 for Large Language Model Applications. https://genai.owasp.org/llm-top-10/ (verified 18 November 2026)
2. NIST. SP 800-218A, Secure Software Development Practices for Generative AI and Dual-Use Foundation Models. https://csrc.nist.gov/pubs/sp/800/218/a/final (verified 18 November 2026)
3. CISA. Known Exploited Vulnerabilities Catalog. https://www.cisa.gov/known-exploited-vulnerabilities-catalog (verified 18 November 2026)
4. NIST National Vulnerability Database. Vulnerabilities API. https://nvd.nist.gov/developers/vulnerabilities (verified 18 November 2026)
5. ProjectDiscovery. Nuclei overview. https://docs.projectdiscovery.io/tools/nuclei/overview (verified 18 November 2026)
6. OISF. Suricata 7.0 User Guide, HTTP keywords. https://docs.suricata.io/en/suricata-7.0.0/rules/http-keywords.html (verified 18 November 2026)
7. Salesforce Engineering. Easily Identify Malicious Servers on the Internet with JARM. https://engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a/ (verified 18 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

Appendix A lists the indicators published for this study. Appendix B carries every published detection rule in full.

INSIDE

A. Indicators of compromise (17) 34

B. Detection rules (4 files) 35

A Indicators of Compromise

Values are defanged for display. The CSV, JSON and STIX files on the online edition carry the raw values. Values the report marks as withheld are not listed.

ID	Type	Indicator	Port	Context	Action
R20-IOC-01	IPv4	107.175.69[.]167	8000	autocve	BLOCK
R20-IOC-02	IPv4	150.158.23[.]244	8000	autocve	BLOCK
R20-IOC-03	IPv4	150.158.93[.]134	8000	autocve	BLOCK
R20-IOC-04	IPv4	43.130.10[.]7	8000	autocve	BLOCK
R20-IOC-05	IPv4	43.159.59[.]133	8000	autocve	BLOCK
R20-IOC-06	IPv4	49.232.241[.]96	8000	autocve	BLOCK
R20-IOC-07	IPv4	8.147.132[.]181	8000	autocve	BLOCK
R20-IOC-08	IPv4	109.122.18[.]186	8084	strix	BLOCK
R20-IOC-09	IPv4	109.122.18[.]186	8085	strix	BLOCK
R20-IOC-10	IPv4	109.122.18[.]186	8086	strix	BLOCK
R20-IOC-11	IPv4	109.122.18[.]186	8089	strix	BLOCK
R20-IOC-12	IPv4	109.122.18[.]186	8093	strix	BLOCK
R20-IOC-13	IPv4	45.207.222[.]140	80	autohunter	BLOCK
R20-IOC-14	IPv4	8.217.238[.]191	8000	cairn	BLOCK
R20-IOC-15	IPv4	147.224.215[.]112	80	redamon	BLOCK
R20-IOC-16	IPv4	169.58.136[.]189	80	redamon	BLOCK
R20-IOC-17	IPv4	23.111.14[.]152	80	redamon	BLOCK

B Detection Rules

Each rule file loads in its own engine (YARA-X, Suricata 7, Kusto, pySigma). Tune thresholds and allow-lists to your environment before alerting or blocking. The same files can be downloaded from the online edition.

odin_deepaudit_autocve_cairn_strix_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Monograph 20 / 30 YARA Artifact Suite
   Scope: Sanitized configuration and orchestration manifests for DeepAudit,
          AutoCVE, Cairn/AutoHunter, and Strix/RedAmon/XingRin stacks.
   Classification: TLP:CLEAR
*/

rule ODIN_DeepAudit_MultiAgent_AST_Config
{
    meta:
        author = "Odin by Brandefense"
        monograph = "20 / 30"
        classification = "TLP:CLEAR"
        usage = "dual-use policy review"
        description = "Matches DeepAudit multi-agent AST sink-to-source configuration manifests"
    strings:
        $id1 = "DEEPAUDIT_" ascii
        $id2 = "deepaudit_backend" ascii nocase
        $ast1 = "sink_to_source" ascii nocase
        $ast2 = "tree_sitter_languages" ascii nocase
        $api1 = "/api/v1/audit" ascii
    condition:
        filesize < 512KB and ($id1 or $id2) and ($ast1 or $ast2) and $api1
}

rule ODIN_AutoCVE_PatchDiff_Synthesizer_Manifest
{
    meta:
        author = "Odin by Brandefense"
        monograph = "20 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt and analyst review"
        description = "Matches AutoCVE advisory ingestion and patch-diff worker orchestration configs"
    strings:
        $ac1 = "AUTOCVE_ENGINE" ascii
        $ac2 = "autocve_task" ascii nocase
        $df1 = "patch_diff_prompt" ascii nocase
        $df2 = "nvd_feed_sync" ascii nocase
        $vr1 = "template_status" ascii nocase
    condition:
        filesize < 512KB and ($ac1 or $ac2) and ($df1 or $df2) and $vr1
}

rule ODIN_Cairn_AutoHunter_Recon_Queue_Config
{
    meta:
        author = "Odin by Brandefense"
        monograph = "20 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt and analyst review"
        description = "Matches Cairn (:8000/:8010) and AutoHunter (:18800) autonomous surface discovery configs"
    strings:
        $c1 = "X-Cairn-Worker" ascii
        $c2 = "/api/v1/cairn/" ascii
        $a1 = "/api/autohunter/" ascii
        $a2 = "18800" ascii
        $q1 = "target_scope" ascii nocase
    condition:
        filesize < 512KB and (($c1 and $c2) or ($a1 and $a2)) and $q1
}
```

```
rule ODIN_Strix_RedAmon_XingRin_Swarm_Compose
{
    meta:
        author = "Odin by Brandefense"
        monograph = "20 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt and analyst review"
        description = "Matches containerized Strix, RedAmon, or XingRin validation swarm manifests"
    strings:
        $s1 = "/api/strix/containers" ascii
        $s2 = "sandbox_port" ascii
        $r1 = "RedAmon-Agent/" ascii
        $r2 = "/api/redamon/" ascii
        $x1 = "/api/xingrin/node" ascii
        $x2 = "X-XingRin-Token" ascii
    condition:
        filesize < 512KB and (($s1 and $s2) or ($r1 and $r2) or ($x1 and $x2))
}
```

odin_specialized_ai_vuln_agents.rules

SURICATA

```
# Odin by Brandefense - Monograph 20 / 30 Suricata 7.x Detection Suite
# Coverage: AutoCVE, DeepAudit (policy review), Cairn, AutoHunter, Strix, RedAmon, XingRin
# Classification: TLP:CLEAR

alert http $HOME_NET any -> $EXTERNAL_NET [8000,3000,13000] (msg:"ODIN [R20] AutoCVE N-Day Advisory Patch-Diff
Synthesis Task Dispatch"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/v1/
cve/"; fast_pattern; http.request_body; bsize:>28; content:"\"cve_id\""; content:"\"patch_diff\""; distance:0;
classtype:trojan-activity; sid:9520001; rev:1; metadata:created_at 2026_11_18, deployment perimeter, tactic
initial_access;)

alert http $EXTERNAL_NET [8000,3000] -> $HOME_NET any (msg:"ODIN [R20] AutoCVE Verification Controller Task
Status Response"; flow:established,from_server; http.stat_code; content:"200"; http.response_body; bsize:>32;
content:"\"autocve_task\""; fast_pattern; content:"\"template_status\""; distance:0; classtype:trojan-activity;
sid:9520002; rev:1; metadata:created_at 2026_11_18, deployment perimeter, tactic reconnaissance;)

alert http $HOME_NET any -> $EXTERNAL_NET [8000,8001,3000,3500,8888] (msg:"ODIN [R20] POLICY Dual-Use DeepAudit
AST Sink-to-Source Audit API"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/
v1/audit"; fast_pattern; http.request_body; bsize:>24; content:"\"repo_source\""; content:"\"sink_trace\"";
distance:0; classtype:policy-violation; sid:9520003; rev:1; metadata:created_at 2026_11_18, deployment perimeter,
policy dual_use_review;)

alert http $HOME_NET any -> $EXTERNAL_NET [8000,8010] (msg:"ODIN [R20] Cairn Autonomous Attack-Surface Recon
Worker Check-In"; flow:established,to_server; http.uri; content:"/api/v1/cairn/"; fast_pattern; http.header;
bsize:>18; content:"X-Cairn-Worker:"; classtype:attempted-recon; sid:9520004; rev:1; metadata:created_at
2026_11_18, deployment perimeter, tactic reconnaissance;)

alert http $HOME_NET any -> $EXTERNAL_NET [18800,18081,80,8080] (msg:"ODIN [R20] AutoHunter Parameter-Mining
Recon Queue Dispatch"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/
autohunter/"; fast_pattern; http.request_body; bsize:>22; content:"\"target_scope\""; classtype:attempted-recon;
sid:9520005; rev:1; metadata:created_at 2026_11_18, deployment perimeter, tactic reconnaissance;)

alert http $HOME_NET any -> $EXTERNAL_NET [8080,8084:8116] (msg:"ODIN [R20] Strix Containerized Validation Swarm
Sandbox Control"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/strix/
containers"; fast_pattern; http.request_body; bsize:>24; content:"\"sandbox_port\""; classtype:trojan-activity;
sid:9520006; rev:1; metadata:created_at 2026_11_18, deployment perimeter, tactic initial_access;)

alert http $HOME_NET any -> $EXTERNAL_NET [80,443,3000,7777] (msg:"ODIN [R20] RedAmon Autonomous Web and API
Validation Agent Beacon"; flow:established,to_server; http.uri; content:"/api/redamon/"; fast_pattern;
http.user_agent; bsize:>14; content:"RedAmon-Agent/"; startswith; classtype:trojan-activity; sid:9520007; rev:1;
metadata:created_at 2026_11_18, deployment perimeter, tactic initial_access;)

alert http $HOME_NET any -> $EXTERNAL_NET [8083,8084,80] (msg:"ODIN [R20] XingRin Distributed Scanner Node
Heartbeat"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/xingrin/node";
fast_pattern; http.header; bsize:>16; content:"X-XingRin-Token:"; classtype:attempted-recon; sid:9520008; rev:1;
metadata:created_at 2026_11_18, deployment perimeter, tactic reconnaissance;)
```

sentinel_autocve_cairn_and_ai_scanner_burst.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 20 / 30 Microsoft Sentinel & Defender XDR Hunt
// Correlates egress to confirmed AI agent controllers and unapproved AI scanner burst activity
// Classification: TLP:CLEAR
// The host allowlist applies only to the port and process branches. A connection to a
// confirmed controller address always alerts, whatever the device name.

let Lookback = 7d;
let ApprovedAppSecHosts = dynamic(["APPSEC-CI-01", "DEVSECOPS-SCAN-01", "VULN-LAB-RUNNER-01"]);
let ConfirmedAgentControllerIPs = dynamic([
    "150.158.23.244", "49.232.241.96", "107.175.69.167", "150.158.93.134",
    "43.130.10.7", "43.159.59.133", "8.147.132.181", "45.207.222.140",
    "8.217.238.191", "147.224.215.112", "169.58.136.189", "23.111.14.152",
    "109.122.18.186"
]);
let AgentControlPorts = dynamic([3000, 3500, 8000, 8010, 8080, 8083, 8084, 8085, 8086, 8089, 8093, 18800]);
let UnapprovedAgentProcs = DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAppSecHosts))
    | where ProcessCommandLine has_any ("autocve", "deepaudit", "cairn-worker", "autohunter", "strix-agent",
"redamon", "xingrin")
        or (FileName in~ ("python", "python3", "uvicorn", "node", "docker") and ProcessCommandLine has_any
("cve_synthesizer", "sink_trace", "18800", "strix"))
    | project ProcTime = Timestamp, DeviceId, DeviceName, InitiatingProcessFileName, FileName,
ProcessCommandLine;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback)
| where RemoteIP in (ConfirmedAgentControllerIPs)
    or (not(DeviceName in~ (ApprovedAppSecHosts))
        and RemotePort in (AgentControlPorts)
        and InitiatingProcessFileName in~ ("python", "python3", "node", "nuclei", "httpx", "curl"))
| summarize
    NetFirstSeen = min(Timestamp),
    NetLastSeen = max(Timestamp),
    DistinctRemoteIPs = dcount(RemoteIP),
    DistinctPorts = make_set(RemotePort, 16),
    MatchedControllerIPs = make_set_if(RemoteIP, RemoteIP in (ConfirmedAgentControllerIPs), 16),
    ConnectionCount = count()
    by DeviceId, DeviceName, InitiatingProcessFileName
| join kind=leftouter (UnapprovedAgentProcs) on DeviceId, DeviceName
| where array_length(MatchedControllerIPs) > 0 or isnotempty(ProcessCommandLine) or DistinctRemoteIPs >= 15
| extend AlertAction = case(
    array_length(MatchedControllerIPs) > 0, "BLOCK_CONFIRMED_AI_AGENT_CONTROLLER",
    ProcessCommandLine has "deepaudit", "REVIEW_UNAPPROVED_DUAL_USE_DEEPAUDIT",
    "HUNT_HIGH_RATE_AI_SCANNER_SWARM"
)
| project NetFirstSeen, NetLastSeen, DeviceName, AlertAction, MatchedControllerIPs, DistinctRemoteIPs,
DistinctPorts, InitiatingProcessFileName, ProcessCommandLine, ConnectionCount
| sort by ConnectionCount desc
```

sigma_specialized_ai_exploit_agent_execution.yml

SIGMA

```
title: Unapproved AutoCVE Advisory Patch-Diff and Template Runner Execution
id: 6d9b4e10-8f2a-4c73-9a51-202611182001
status: experimental
description: Detects unapproved execution of AutoCVE LLM advisory patch-diff synthesis workers or automated
template verification pipelines on non-AppSec hosts.
author: Odin by Brandefense
date: 2026/11/18
tags:
  - attack.initial_access
  - attack.resource_development
logsource:
  category: process_creation
  product: linux
```

```
  detection:
    selection_runtime:
      Image|endswith:
        - '/python'
        - '/python3'
        - '/uvicorn'
        - '/nuclei'
    selection_cli:
      CommandLine|contains:
        - 'autocve'
        - 'cve_synthesizer'
        - 'patch_diff_agent'
        - '--cve-feed'
    filter_approved_runners:
      ComputerName:
        - 'APPSEC-CI-01'
        - 'DEVSECOPS-SCAN-01'
    condition: selection_runtime and selection_cli and not filter_approved_runners
  falsepositives:
    - Authorized vulnerability management runners listed in asset inventory
  level: high
  ---
  title: Unapproved Dual-Use DeepAudit AST Sink-to-Source Repository Mining
  id: 7a3c15e2-51b8-48d9-b044-202611182002
  status: experimental
  description: Identifies unapproved DeepAudit multi-agent LLM code-auditing backends launched outside sanctioned
  application security pipelines.
  author: Odin by Brandefense
  date: 2026/11/18
  tags:
    - attack.reconnaissance
    - attack.resource_development
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_proc:
      Image|endswith:
        - '/python3'
        - '/uvicorn'
        - '/node'
    selection_args:
      CommandLine|contains:
        - 'deepaudit'
        - 'sink_to_source'
        - 'ast_audit_worker'
    filter_appsec:
      ComputerName:
        - 'APPSEC-CI-01'
        - 'DEVSECOPS-SCAN-01'
    condition: selection_proc and selection_args and not filter_appsec
  falsepositives:
    - Sanctioned DevSecOps static analysis servers; apply dual-use watchlist review before host isolation
  level: medium
  ---
  title: Containerized Cairn AutoHunter Strix RedAmon or XingRin Recon Swarm Spawn
  id: 9f402d8b-3e19-46a2-8c17-202611182003
  status: experimental
  description: Detects worker processes or container sandboxes associated with Cairn, AutoHunter (:18800), Strix (:
  8084-:8116), RedAmon, or XingRin (:8083) autonomous scanning agents.
  author: Odin by Brandefense
  date: 2026/11/18
  tags:
    - attack.reconnaissance
    - attack.initial_access
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_agent:
      CommandLine|re: '(cairn[-_]worker|autohunter.*18800|strix[-_](agent|sandbox)|redamon[-_]worker|
  xingrin.*8083)'
```

```
    filter_lab:
      ComputerName:
        - 'VULN-LAB-RUNNER-01'
    condition: selection_agent and not filter_lab
  falsepositives:
    - Dedicated red-team validation sandboxes registered with security operations
  level: high
```

ODIN

About Odin Research

Odin Research is the public research series of Odin by Brandefense. It documents adversary infrastructure seen on the internet and turns each finding into indicators and detection rules that defenders can use. Victims and third parties stay anonymous, addresses appear only for confirmed adversary infrastructure, and no working attack code is published.

Odin is the agentic analyst inside the Brandefense threat intelligence platform, which unifies Cyber Threat Intelligence, Digital Risk Protection and External Attack Surface Management.

Online edition, indicators and rules · brandefense.io/odin-research/reports/specialized-ai-vulnerability-agents/
Short Read · brandefense.io/odin-research/blog/ai-agents-now-specialize-in-vulnerability-work-one-step/
Defense Kit · brandefense.io/odin-research/defense-kit/