ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 53 / 30 · OPERATIONS

Operation LedgerSwarm

One Operator's AI Agent Campaign Against Payments and Web3

Inside one operator's 76-organization swarm archive, 10.8 billion input tokens across 39 tasks, and defenses for fintech and digital assets

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

76
Anonymized target organizations across 18 verticals in the exposed campaign archive

39
Partially captured task records carrying full timing, sector, country, template, and token metadata

6,124
Unverified LLM-labeled findings across 141 task IDs plus 85 unassigned (941 critical, 2,173 high)

61 + 174
Dedicated follow-up scripts (61) plus swarm-tuning files (126) and prompt/skill documents (48)

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

Inside one operator's 76-organization swarm archive, 10.8 billion input tokens across 39 tasks, and defenses for fintech and digital assets

01	Executive Summary	04
02	One Campaign Behind a Swedish Relay	07
03	Campaign Timeline, Token Scale, and the Scope-Lock Shift	10
04	From Swarm Sweeps to 61 Custom Follow-Up Scripts	14
05	Why Payments, Web3, and Exchanges Lead	17
06	Weighing the Authorized Red-Team Hypothesis	20
07	What Defenders Should Do	23
08	Detection Rules	25
09	Indicators	28
10	Scope and Limits	30
11	Sources	32
A	Indicators of compromise (2)	34
B	Detection rules (4 files)	35

IN SHORT

Operation LedgerSwarm tracks one operator's autonomous agent campaign across 76 target organizations, 10.8 billion input tokens across 39 tasks, and 61 follow-up scripts. Odin by Brandefense analyzed the exposed campaign archive in a two-week observation window ending 18 November 2026. Payments, Web3, banking, and exchange teams must remove front-end secrets, adopt short-lived cloud credentials, lock bucket writes to deployment accounts, and alert on 25-URI scan bursts.

01

CHAPTER 01

Executive Summary

IN SHORT

Operation LedgerSwarm tracks one operator's autonomous agent campaign across 76 target organizations, 10.8 billion input tokens across 39 tasks, and 61 follow-up scripts. Odin by Brandefense analyzed the exposed campaign archive in a two-week observation window ending 18 November 2026. Payments, Web3, banking, and exchange teams must remove front-end secrets, adopt short-lived cloud credentials, lock bucket writes to deployment accounts, and alert on 25-URI scan bursts.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Autonomous offense frameworks place a large language model (LLM) in charge of discovery and vulnerability tools. One operator can queue dozens of organizations and let parallel subagents enumerate subdomains, check JavaScript bundles, and draft summaries continuously.

Operation LedgerSwarm isolates one operator's campaign behind a virtual private server (VPS) in Sweden acting as a relay in front of a second server. Dates before our November window come from the operator's own task records. All target names, domains, task names, archive paths, and both server addresses are withheld.

1. **Payments and Web3 firms make up 53.9% of targets and 85.5% of input tokens.**
   Across 76 target organizations in 18 verticals, Payments and Fintech (21) and Web3, DeFi, and Stablecoin (20) lead. Within 39 captured task records carrying metadata, those two sectors account for 29 tasks and 9.23 billion of 10.79 billion input tokens.

2. **The operator shifted to strict domain-locked templates before writing 61 custom scripts.**
   Seven early tasks carry dates from 9 to 12 October 2026, led by one 22-day British fintech run (`3.44 billion` input tokens). After 23 pre-lock tasks from 4 to 9 November, templates on 9 and 10 November locked workers to whitelisted domains and blocked neighbouring-address scans.

3. **Swarm reports claim 6,124 unverified findings while custom scripts target cloud keys.**
   Across 141 task IDs plus 85 unassigned entries, the swarm's own summaries record 6,124 claimed findings (941 critical, 2,173 high). None is a confirmed breach. Where automated scans flagged leads, the operator wrote 61 dedicated follow-up scripts, including the Google Cloud Storage playbook in Monograph 30.

Three Decisions for Leaders

1. **Remove front-end secrets and static workstation cloud keys now.** Strip API keys and `.env` files from web roots, JavaScript bundles, and repositories. Move developers from long-lived service-account JSON keys to short-lived federated tokens.
2. **Restrict object storage writes to named deployment identities.** Treat any Cloud Storage, S3, or R2 bucket that feeds a web root or CDN as production code. Disable PHP execution on media mounts and alert on non-CI writes.
3. **Throttle automated discovery bursts across APIs and subdomains.** Pair external attack surface management with per-source API rate limits, bot detection, and disabled production GraphQL introspection. Alert when one source requests 25 or more distinct URIs in five minutes.

The Study in Numbers

Number	What it is
76	Anonymized target organizations across 18 verticals in the exposed campaign archive
13,864 / 1,105	Subdomains enumerated across those 76 targets, narrowed to 1,105 live HTTP or TLS FQDNs
39	Partially captured task records carrying full timing, sector, country, template, and token metadata
10,794,890,273	LLM input tokens consumed across the 39 metadata tasks, plus 132,372,680 output tokens
6,124	Unverified LLM-labeled findings across 141 task IDs plus 85 unassigned (941 critical, 2,173 high)
61 + 174	Dedicated follow-up scripts (61) plus swarm-tuning files (126) and prompt/skill documents (48)
2	Published indicators in Appendix A: 2 follow-up script SHA-256 hashes and 0 addresses

How to use this report.

Security leaders can focus on Chapters 1, 5, 6, and 7. Cloud, DevOps, and application teams in payments, Web3, and exchanges should read Chapters 4, 5, and 7. SOC analysts and detection engineers should start with Chapters 3, 8, and 9. Threat intelligence analysts will find architecture, hypotheses, and limits in Chapters 2, 6, and 10.

02

CHAPTER 02

One Campaign Behind a Swedish Relay

IN SHORT

ARTEX is a public autonomous penetration-testing framework used by many unrelated groups. Operation LedgerSwarm examines one operator's campaign behind a Swedish VPS relay fronting a second server, spanning 76 target organizations, 39 metadata tasks, and 235 custom files.

INSIDE THIS CHAPTER

The two-server relay and what is withheld 07

How the archive views fit together 08

02 One Campaign Behind a Swedish Relay

On 13 November 2026, the FBI issued advisory `FLASH-20261113-001` warning that threat actors worldwide had weaponized ARTEX. Built in Go with a Next.js console, ARTEX connects external language models to automated discovery and exploitation tools. Earlier in November 2026, Censys documented 1,465 hosts running exposed ARTEX consoles, often on port 8787; defenders can assess such exposure with external attack surface management.

In October 2026, Microsoft Threat Intelligence reported that a China-linked actor (`Storm-0940`) used ARTEX during cloud supply-chain intrusions. Those public disclosures provide **framework context only**. ARTEX is a shared tool used by many unrelated operators, and nothing links `Storm-0940` or other named groups to Operation LedgerSwarm.

Monograph 11 mapped 4,360 endpoints across five AI offense frameworks (2,121 ARTEX endpoints, 495 confirmed) and introduced the 76-organization archive. Monograph 30 analyzed two Python scripts from that archive written to plant a `pax_cmd` WordPress gate through Google Cloud Storage. Monograph 53 examines the operation end to end, translating its timeline and token scale into cyber threat intelligence priorities. Figure 1 maps the four task waves and workspace layers.

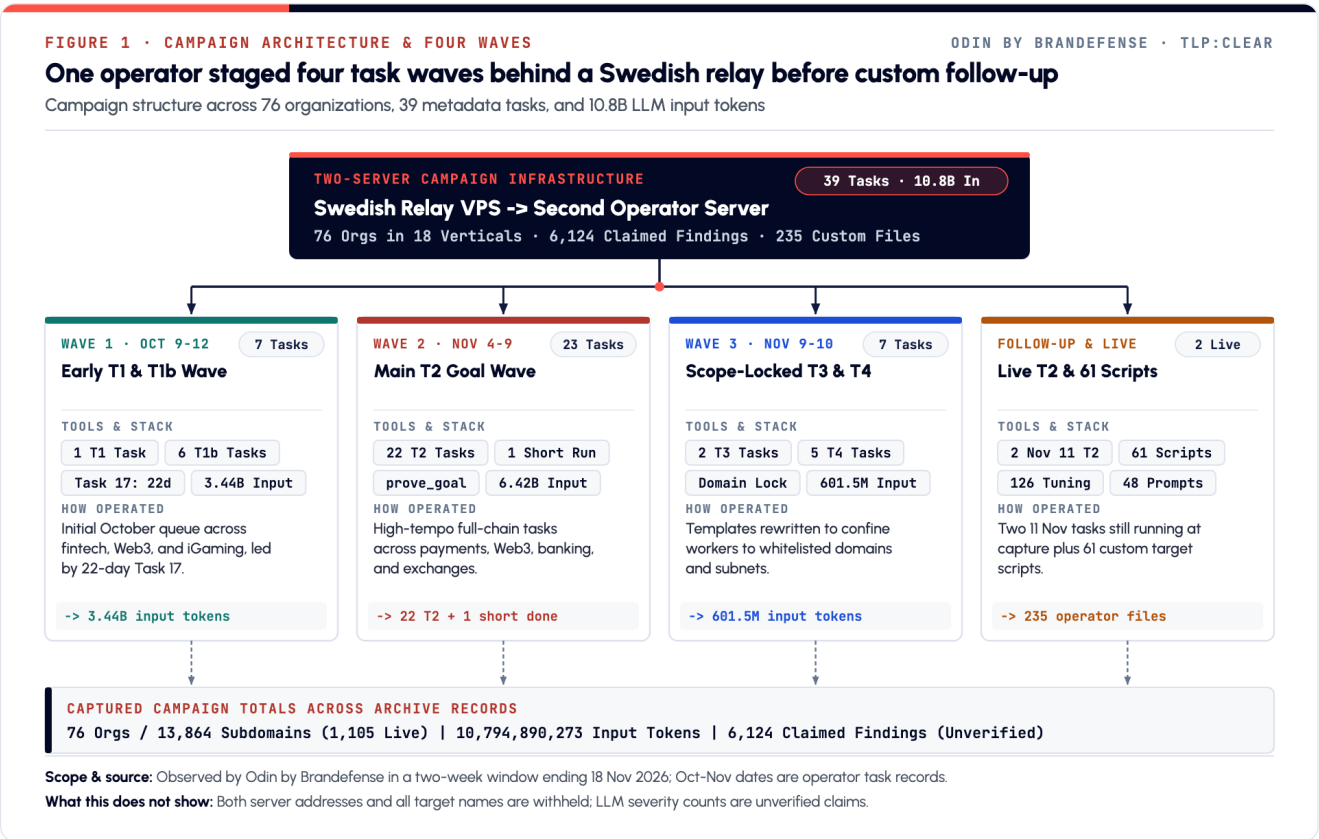


Figure 1: One operator staged four task waves behind a Swedish relay before custom follow-up

The two-server relay and what is withheld

The operator placed **a VPS in Sweden acting as a relay in front of a second server**. The Swedish host forwards traffic to the upstream controller and also exposes the campaign archive. Because that archive still holds target data, both server IP addresses and every archive path are withheld.

Every target identity is withheld under the same rule. No target organization name, domain, cloud bucket, WordPress theme, or task name appears in this report, its figures, or its rules. Individual tasks are referenced only by anonymized numeric IDs (8 through 159).

How the archive views fit together

Because the task list was captured partially during our two-week window ending 18 November 2026, the archive provides distinct views of the campaign. Keeping them separate avoids double-counting.

Archive view	Scope	Key counts	Main limit
Organization roll-up (M11)	External surface	76 orgs in 18 verticals; 13,864 subdomains (1,105 live); 35 crit/high scopes (17 crit, 18 high)	Unverified LLM scope labels
Task metadata slice	Captured task rows	39 tasks (37 `done`, 2 `running`); 10,794,890,273 input & 132,372,680 output tokens	Partial capture (8–19, 125–159)
Per-task claimed findings	Per-task counters	6,124 claims across 141 task IDs (6,039) plus 85 unassigned	Unverified tool and LLM labels
Anchor worker runs (M30)	Deep-task effort	2,178 runs (crypto exchange); 1,231 runs (iGaming)	Effort only; impact unconfirmed
Operator workspace	Custom files	61 follow-up scripts, 126 tuning files, 48 prompt/skill docs (235 total)	Counts only; code & prompts withheld

CHAPTER 03

Campaign Timeline, Token Scale, and the Scope-Lock Shift

IN SHORT

Task records inside the 39-task slice carry operator creation dates from 9 October to 11 November 2026 and total 10.79 billion input tokens. On 9 and 10 November, the operator updated templates to lock workers to whitelisted domains and block neighbouring-address scans.

INSIDE THIS CHAPTER

Four waves in the operator's task records 11

Three tasks burned 58.7% of all input tokens 11

Self-graded goals do not confirm breaches 11

The 9 and 10 November domain and subnet lock 12

03 Campaign Timeline, Token Scale, and the Scope-Lock Shift

Dates earlier than November 2026 in this chapter are **task creation and completion dates from the operator's own controller logs**, not Odin monitoring dates. Together, the 39 metadata tasks consumed **10,794,890,273 LLM input tokens** and **132,372,680 output tokens** (10,927,262,953 tokens combined).

Input tokens outnumber output tokens by **81.5 to 1**. Worker pools feed HTTP responses, JavaScript bundles, API schemas, and tool logs into the model on every step while receiving compact JSON calls back. Figure 2 links the four template generations and custom files to defender controls.

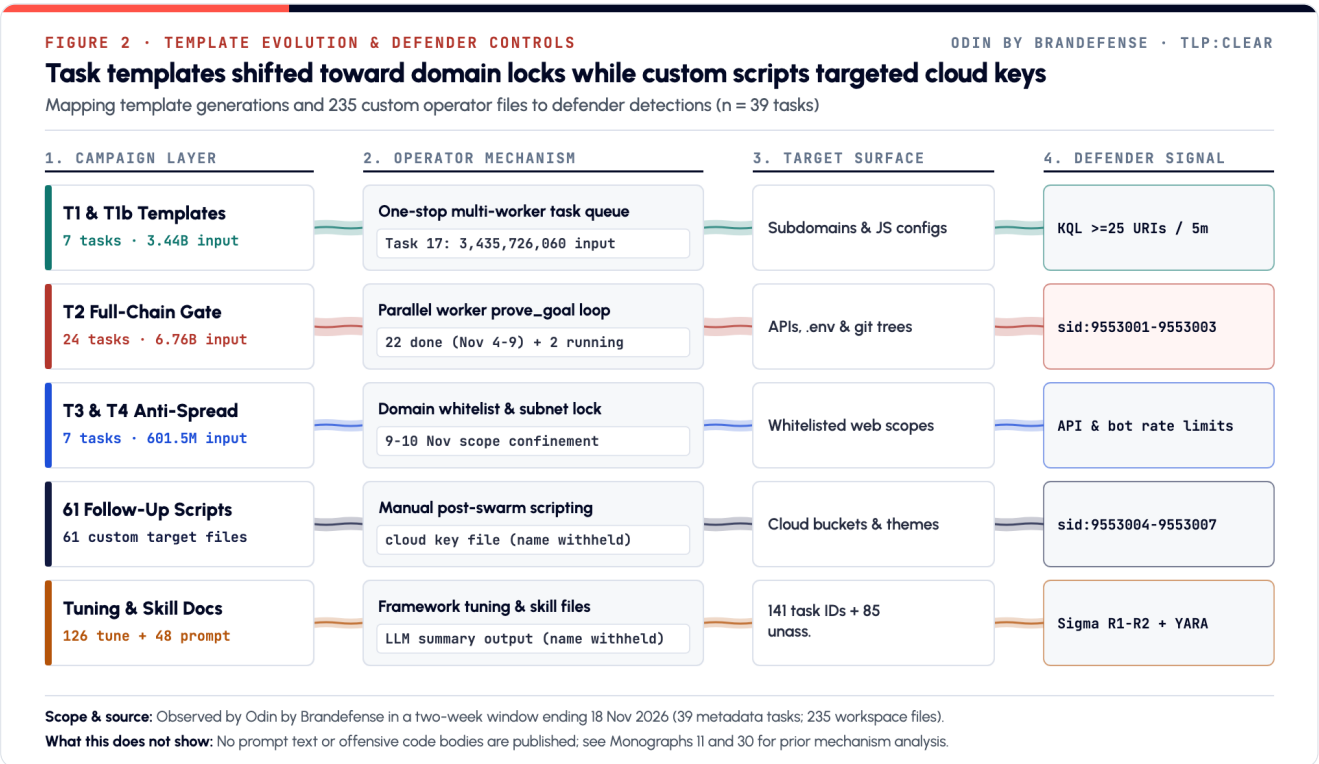


Figure 2: Task templates shifted toward domain locks while custom scripts targeted cloud keys

Four waves in the operator's task records

Grouping the 39 metadata tasks by creation date and template generation shows how the campaign progressed.

Phase and template	Operator record dates	Tasks	Input tokens	Output tokens	Claimed crit / high	Status and goals
First wave: `T1` (1) and `T1b` (6), incl. Task 17	9 Oct to 3 Nov 2026	7	3,435,752,673	27,095,532	149 / 450	7 `done` (2 of 7 goals)
Main wave: `T2` full-chain (`prove_goal` gate)	4 to 10 Nov 2026	22	6,418,031,562	91,051,603	284 / 434	22 `done` (60 of 60 goals)
Main wave short run: `unnamed` template	5 Nov 2026	1	46,773	1,051	0 / 0	1 `done` (0 of 0 goals)
Scope-restricted wave: `T3` hard scope lock	9 Nov 2026	2	64,778,721	1,013,809	7 / 2	2 `done` (5 of 5 goals)
Scope-restricted wave: `T4` v7 subnet lock	10 Nov 2026	5	536,674,733	8,174,897	10 / 39	5 `done` (11 of 11 goals)
Active at capture: `T2` full-chain (`prove_goal`)	Created 11 Nov 2026	2	339,605,811	5,035,788	7 / 20	2 `running` (2 of 2 goals)
Total (39-task metadata slice)	**9 Oct to 11 Nov 2026**	**39**	**10,794,890,273**	**132,372,680**	**457 / 945**	**37 `done`, 2 `running` (80/85)**

Across the full table, **24 tasks use the `T2` template** (6,757,637,373 input tokens, 62.6%). Twenty-two completed in the 4 to 9 November wave, and 2 created on 11 November remained `running`. Adding 1 five-second `unnamed` run (Task 137 on 5 November) gives 23 pre-lock tasks in that wave. Ten sampled session report dates also cluster between 2 and 9 November (1 on 2 Nov, 3 on 3 Nov, 5 on 5 Nov, 1 on 9 Nov).

Three tasks burned 58.7% of all input tokens

Token spend was concentrated in long multi-day runs. Two early October tasks (Tasks 8 and 13) logged zero tokens in metadata. By contrast, **Task 17** (Payments and fintech, United Kingdom) spanned 22 days from 12 October to 3 November 2026. It consumed **3,435,726,060 input tokens** (31.8% of the 39-task total) and 27,093,427 output tokens across 566 claimed findings.

Two `T2` tasks followed in November. **Task 145** (`NL` payments, 7 to 10 November) used 1,566,410,200 input tokens. **Task 133** (`US` Web3, 5 to 7 November) used 1,335,524,890 input tokens. Together, Tasks 17, 145, and 133 consumed **6,337,661,150 input tokens (58.7%)**.

Self-graded goals do not confirm breaches

Starting with `T2`, the operator added a `prove_goal` check. Across all 31 tasks using `T2`, `T3`, or `T4`, the controller marked **78 of 78 defined goals met (100%)**. Yet Tasks 128 and 144 both recorded 100% goals met while logging zero critical and zero high findings. A `prove_goal` flag only marks checklist completion inside the agent loop, not a verified target intrusion.

The 9 and 10 November domain and subnet lock

On 9 November (`T3`, Tasks 151 and 152) and 10 November (`T4` v7, Tasks 153 to 157), the operator rewrote the task template. Both versions **confine each task to whitelisted target domains and forbid touching neighbouring addresses**.

Average input token use dropped to 32.4 million per task under `T3` and 107.3 million under `T4`, compared with 281.6 million across `T2`. We separate observation from interpretation:

- **Observed fact.** `T3` and `T4` confine subagents to whitelisted domains and forbid neighbouring-address scans, before Tasks 158 and 159 returned to `T2` on 11 November.
- **Hypotheses.** Generating less network noise and hitting fewer unintended third-party hosts on shared subnets are plausible motives, though the archive does not state why the rule was added.

04

CHAPTER 04

From Swarm Sweeps to 61 Custom Follow-Up Scripts

IN SHORT

The swarm claimed 6,124 findings across 141 task IDs plus 85 unassigned entries, all written by an LLM. The archive also holds 61 follow-up scripts, 126 tuning files, and 48 prompt and skill documents. Treat the scripts as staged intent until target-side activity is verified.

INSIDE THIS CHAPTER

Reading the 6,124 claimed findings 14

235 custom operator files 15

04 From Swarm Sweeps to 61 Custom Follow-Up Scripts

Operation LedgerSwarm shows how an operator pairs automation with manual scripting. The swarm sweeps dozens of scopes first; the human operator then writes custom scripts for targets that expose keys or cloud paths. See Monograph 11 for ARTEX's four-pool worker architecture, and see Monograph 30 for the Cloud Storage-to-WordPress chain. Figure 3 models that handoff from automated discovery to cloud follow-up.

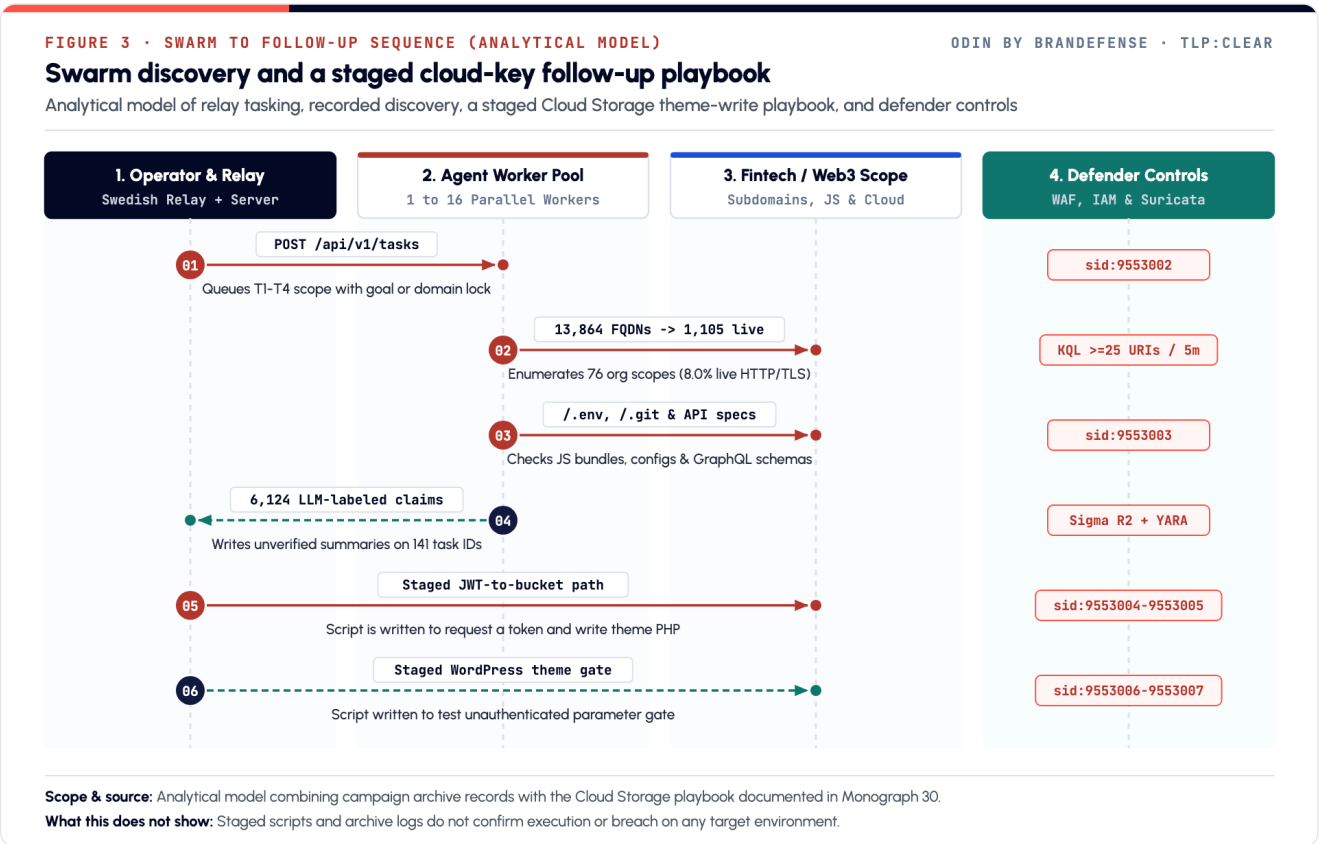


Figure 3: Recorded subdomain discovery and a staged cloud-key follow-up playbook

Reading the 6,124 claimed findings

Across the per-task summary table, the swarm logged **6,124 claimed findings**: 6,039 across 141 numbered task IDs (8 through 159) plus 85 unassigned entries.

Claimed severity	Count	Share of 6,124	How defenders should read it
Critical	941	15.4%	LLM label; often flags `.env` hits, API docs, or unverified banners
High	2,173	35.5%	LLM label; includes open admin panels, GraphQL schemas, and config leaks
Medium	2,161	35.3%	Automated header and path findings; unverified on target hosts
Low	837	13.7%	Informational banners and minor TLS or cookie settings
Informational / other	12	0.2%	Uncategorized items across seven task rows
Total	**6,124**	**100.0%**	**Unverified swarm claims across 141 task IDs + 85 unassigned**

Fifteen task IDs claim 100 or more findings each, led by Task 109 (604), Task 17 (566), and Task 133 (456). Seventy-six task IDs claim at least one critical item. Treat all 6,124 entries strictly as a map of what the swarm checks, never as confirmed breaches.

235 custom operator files

Alongside automated logs, the workspace held **235 operator-authored files**. Under Odin publication rules, we publish counts only, with zero prompt text, zero exploit code, and zero target names:

- **61 dedicated follow-up scripts** written after the swarm finished for a small set of targets;
- **126 swarm-tuning files** modifying worker concurrency, tool wrappers, and task queues;
- **48 prompt and skill documents** defining task templates, reporting gates, and scope rules.

Monograph 30 examined two staged follow-up scripts (`deploy_gcs_webshell.py`, 5,251 bytes, and `deploy_gcs_webshell2.py`, 5,480 bytes) tied to the archive's iGaming task (**1,231 recorded worker runs**). Both are written to read a harvested Google Cloud key (filename withheld) and request a one-hour OAuth token with `devstorage.full_control` scope.

They are written to overwrite a WordPress theme's `404.php` and `functions.php` inside a Cloud Storage bucket with an unauthenticated `pax_cmd` gate. Separately, the crypto-exchange task (**2,178 recorded worker runs**) logged unverified claims involving linked-server SQL injection, a NAS webshell, and a workstation cloud key.

05

CHAPTER 05

Why Payments, Web3, and Exchanges Lead

IN SHORT

Payments and Fintech (21 organizations) and Web3, DeFi, and Stablecoin (20 organizations) form 53.9% of the 76 archive targets and 85.5% of input tokens in the 39-task slice. Wide API surfaces, public SDK repositories, and cloud buckets make both sectors prime targets.

INSIDE THIS CHAPTER

Sector comparison across both archive views 18

What an agent swarm looks like to defenders 18

05 Why Payments, Web3, and Exchanges Lead

Payment gateways, fintech APIs, DeFi protocols, stablecoin issuers, and exchanges share three traits that reward automated swarms. First, they expose hundreds of subdomains across partner sandboxes, docs portals, RPC nodes, and regional sites.

Second, web and mobile bundles often leak cloud bucket names, GraphQL endpoints, or API keys. Third, stolen cloud or payment keys grant direct financial access without endpoint malware. Figure 5 compares the 76-organization archive with the 39-task metadata slice.

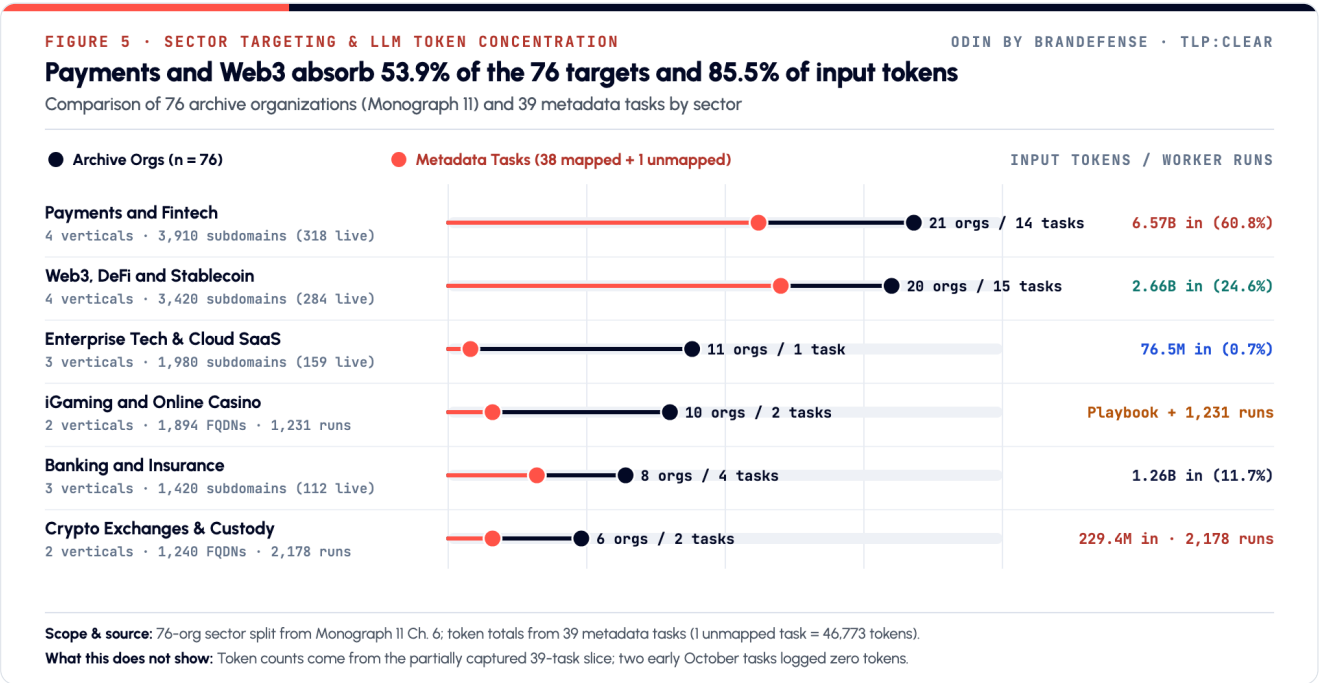


Figure 5: Payments and Web3 absorb 53.9% of the 76 targets and 85.5% of input tokens

Sector comparison across both archive views

Every number in the 76-organization columns matches Monograph 11 Chapter 6, paired with the 39-task metadata slice.

Sector	Verticals	Archive orgs (n=76)	Subdomains / live	Labeled crit or high	Metadata tasks (n=39)	Input tokens (39 tasks)
Payments and Fintech	4	21 (27.6%)	3,910 / 318	10	14 (35.9%)	6,566,351,648 (60.8%)
Web3, DeFi and Stablecoin	4	20 (26.3%)	3,420 / 284	9	15 (38.5%)	2,660,618,441 (24.6%)
Enterprise Tech and Cloud SaaS	3	11 (14.5%)	1,980 / 159	5	1 (2.6%)	76,453,072 (0.7%)
iGaming and Online Casino	2	10 (13.2%)	1,894 / 146	5	2 (5.1%)	7,505 (<0.1%)
Banking and Insurance	3	8 (10.5%)	1,420 / 112	3	4 (10.3%)	1,261,970,830 (11.7%)
Crypto Exchanges and Custody	2	6 (7.9%)	1,240 / 86	3	2 (5.1%)	229,442,004 (2.1%)
Not mapped (5-second Task 137)	0	0 (0.0%)	0 / 0	0	1 (2.6%)	46,773 (<0.1%)
Total	**18**	**76 (100.0%)**	**13,864 / 1,105**	**35 (46.1%)**	**39 (100.0%)**	**10,794,890,273 (100.0%)**

Adding Banking and Insurance (4 tasks) and Crypto Exchanges (2 tasks) pushes financial and digital-asset targets to **35 of 39 tasks (89.7%) and 99.3% of input tokens**. Geographically, the 39 tasks span 15 named countries plus two `XX` rows and one unmapped row. The United States (10), the United Kingdom (8), and India (3) lead, followed by two tasks each in the United Arab Emirates, Germany, and Lithuania.

What an agent swarm looks like to defenders

From the defender's side, an operation like LedgerSwarm leaves four clear signals:

1. **Broad subdomain enumeration.** Across 76 targets, the swarm enumerated 13,864 subdomains (182.4 each) and focused on 1,105 live FQDNs (8.0%).
2. **High-rate URI bursts.** Subagents sweep `/.env`, `/.git/config`, `/graphql`, and OpenAPI paths within minutes, tripping distinct-URI thresholds.
3. **Front-end JavaScript and repo harvesting.** Workers fetch `.js` bundles and exposed configs for cloud keys, database URIs, and API secrets.
4. **Cloud API call pattern.** The staged playbook is written to call `oauth2.googleapis[.]com` and `storage.googleapis[.]com`; hunt for such calls outside approved CI/CD hosts.

CHAPTER 06

Weighing the Authorized Red-Team Hypothesis

IN SHORT

Prompt templates in the workspace describe the tasks as authorized penetration testing. Odin found no evidence of authorization, and the staged cloud-key playbook, a hidden Swedish relay across 14+ countries, and cleanup instructions weigh against a legitimate test.

06 Weighing the Authorized Red-Team Hypothesis

Figure 4 shows the withheld two-server relay layout and the four workspace layers reviewed for intent.

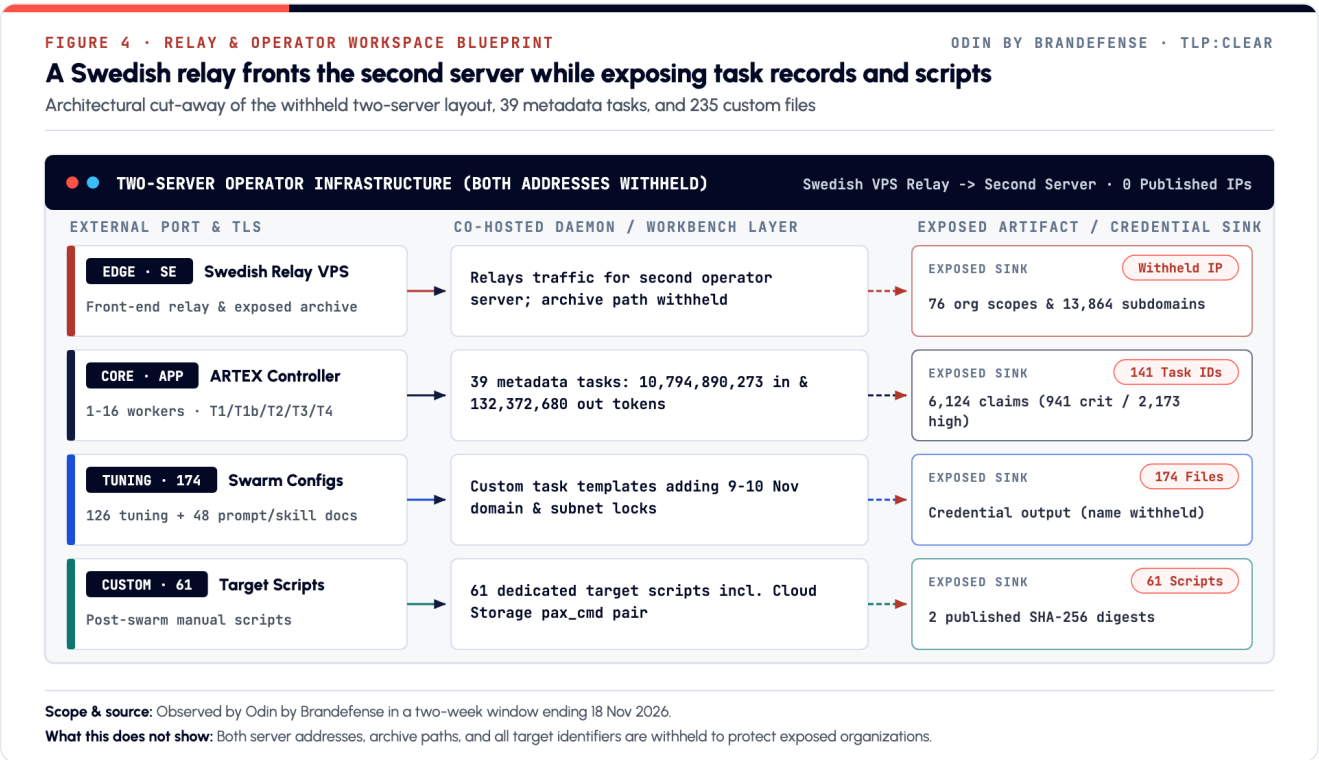


Figure 4: A Swedish relay fronts the second server while exposing task records and scripts

The swarm's prompt files frame its work as **authorized penetration testing**. Operators of offensive AI tools commonly place "authorized assessment" wording in system prompts so commercial or open-weight models do not refuse exploitation tasks.

Odin found **no evidence of authorization**. Treating "authorized red team" as an alternative hypothesis, three high-level facts weigh against it:

1. **Staged stolen-key playbook and unauthenticated gate.** Monograph 30 documents a playbook written to use a harvested Google Cloud key to plant an unauthenticated `pax_cmd` gate in WordPress templates. Target-side execution is unconfirmed.
2. **Hidden two-tier relay and global targeting.** Operating behind a Swedish VPS relay in front of a second server across 76 financial, Web3, and iGaming organizations in 14+ countries does not match a regional consultancy's client list.
3. **Target-side cleanup paired with open data exposure.** Templates include target-side cleanup instructions, while the relay left harvested target records exposed without authentication.

Hypothesis	For	Against	Conclusion
Unapproved single-operator campaign	76 scopes in 14+ countries; Swedish relay; stolen key `pax_cmd` playbook; cleanup rules	Prompt claims "authorized"; `T3` / `T4` domain locks	**Most consistent**
Authorized red team	`T1` – `T4` templates, `prove_goal` gates, 9–10 Nov locks	Zero auth records; staged unauthenticated `pax_cmd` gate; exposed data	**Weighed against**
Shared public server	Public ARTEX consoles exist on VPS networks	Cohesive `T1` – `T4` lineage, 85.5% fintech/ Web3 tokens, 235 files	**Rejected**

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

- [ ] Scan public JavaScript bundles, mobile builds, and repos for API keys, RPC tokens, and `.env` files. Block `/.env` and `/.git/` at the WAF and rotate exposed secrets.
- [ ] Block `pax_cmd` in query strings and POST bodies at the WAF, and disable production GraphQL `__schema` introspection. Verify hashes of deployed `404.php` and `functions.php` files.

Within 72 hours

- [ ] Revoke workstation cloud service-account JSON keys. Enforce `constraints/iam.disableServiceAccountKeyCreation` and use short-lived federated OIDC tokens.
- [ ] Audit every Cloud Storage, S3, or R2 bucket synced into a web root or CDN. Restrict code writes to named CI/CD accounts, turn off PHP on media mounts, and enable `DATA_WRITE` logs.

Within 7 to 30 days

- [ ] Apply per-IP and per-ASN rate limits plus bot detection on `/api/`, `/graphql`, and login routes.
- [ ] Alert when one source IP hits 25 distinct URIs in five minutes, or 8 URIs touching `/.env`, `/.git/config`, or `/graphql`.
- [ ] Reconcile external DNS zones to retire stale staging subdomains and dangling CNAMEs.

08

CHAPTER 08

Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

Suricata, KQL, Sigma, and YARA summary 25

MITRE ATT&CK mapping 26

08 Detection Rules

The full rules are in Appendix B, adapted from Monographs 11 and 30.

Rule file	What it catches
`odin_ledgerswarm_swarm_and_cloud_key_abuse.rules`	Suricata 7 (8 SIDs): ARTEX consoles, `/api/v1/tasks`, `.env` / `.git` sweeps, OAuth `jwt-bearer`, Cloud Storage PHP writes, and `pax_cmd` gates
`sentinel_ledgerswarm_burst_and_cloud_key_hunt.kql`	Sentinel & Defender XDR: 5-minute URI bursts, non-CI Google API calls, and PHP worker shells
`sigma_ledgerswarm_swarm_and_gcs_theme_abuse.yml`	Sigma (2 rules): non-CI Cloud Storage PHP writes and container batch tools
`odin_ledgerswarm_artex_and_gcs_playbook_artifacts.yar`	YARA (3 rules): ARTEX bundles, Cloud Storage deployers, and `pax_cmd` gates

Suricata, KQL, Sigma, and YARA summary

SID	Signal	Purpose
9553001	HTTP 200 body containing `ARTEX` plus a title marker or `artex-swarm-root`	Exposed ARTEX console
9553002	Outbound `POST /api/v1/tasks` (`target_scope`)	Swarm task dispatch
9553003	Five `GET` requests in 60s for `/.env*` or `/.git/config`	Edge secret sweep
9553004	Outbound `POST` to `oauth2.googleapis[.]com` (`jwt-bearer`)	Non-CI token mint hunt
9553005	Outbound `POST /upload/storage/v1/b/` (`application/x-php`)	Cloud Storage PHP write
9553006–9553007	`pax_cmd=` in URI (9553006) or POST body (9553007)	Backdoor gate alert; block separately at the WAF
9553008	Direct request for `/wp-content/themes/*/(404|functions).php`	Direct template probe

The 7-day KQL query bins `W3CIISLog` into 5-minute windows per IP (25 URIs, 10 with tool user agents, or 8 on secret paths). Endpoint branches check `DeviceProcessEvents` and `DeviceNetworkEvents` for PHP worker shells, playbook arguments, and non-CI Google API calls using exact allowlists. Sigma covers non-CI Cloud Storage PHP writes and container batch tools, while YARA enforces file-size caps and multi-string matches.

MITRE ATT&CK mapping

Tactic	Technique	Behavior	Detection
Reconnaissance	T1595.002 Vulnerability Scanning	13,864 subdomains; `.env` / `.git` sweeps	SIDs 9553002–003; KQL; Sigma 2
Defense Evasion	T1078.004 Cloud Accounts	Stolen Google service-account key	SID 9553004; KQL; Sigma 1
Credential Access	T1528 Application Access Token	`jwt-bearer` (`devstorage.full_control`)	SID 9553004; YARA
Persistence	T1505.003 Web Shell	Theme `404.php` / `functions.php` `pax_cmd`	SIDs 9553005–008; Sigma 1; YARA

09

CHAPTER 09

Indicators

IN SHORT

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

09 Indicators

Appendix A holds 2 published indicators: 2 file hashes (SHA-256) and 0 network addresses. Five behavioral and file-name patterns appear in this chapter only.

File (in Appendix A)	Size (B)	SHA-256	Use
`deploy_gcs_webshell.py` (v1)	5,251	`20745360f904db566a373ce39bb05ea52fee24c5f6a404db756e9c5aecb1981e`	Endpoint EDR hunt
`deploy_gcs_webshell2.py` (v2)	5,480	`17c96f2ab239b6fd5b5454aae51301bee2021f0b8d9726c567c48b8ab167fbdc`	Endpoint EDR hunt

Pattern (report only)	Type	Use
`cd3ebd16437d6d3963592f7365dd3980`	ARTEX `/favicon.ico` MD5 (15,406 B)	EASM console hunt
`POST /api/v1/tasks`	ARTEX task API route	Outbound alert (SID 9553002)
`pax_cmd`	Query or POST parameter	WAF block; Suricata alert (SIDs 9553006–007)
`oauth2.googleapis[.]com/token` (`jwt-bearer`)	Cloud API pattern	Watch non-CI callers; never block domain
`devstorage.full_control`	OAuth 2.0 scope	Cloud IAM hunt

Never block `oauth2.googleapis[.]com` or `storage.googleapis[.]com`; monitor unusual callers instead. Withheld items include both operator server addresses, archive paths, all 76 target names and domains, task names, credential fragments, target file hashes, and hashes of the remaining 233 operator files.

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 and task dates.** Findings cover a two-week observation window ending 18 November 2026. Dates between 9 October and 11 November 2026 come from the operator's own task records.
- **Claims are not breaches.** The 6,124 claimed findings (941 critical, 2,173 high) and 35 critical/high scopes come from automated tools and an LLM writer, not confirmed target breaches.
- **Partial task capture.** Thirty-nine captured task rows carry full metadata (`10.79 billion` input tokens), while the per-task findings table spans 141 task IDs plus 85 unassigned items.
- **Staged scripts, not target logs.** The 61 follow-up scripts show operator capability and intent, not confirmed execution on any live site.
- **Framework vs operator.** ARTEX is a public tool used by many unrelated actors, and nothing links Operation LedgerSwarm to `Storm-0940` or other named groups.
- **Rule testing.** Every published rule file loaded without errors in Suricata 7.0.12, YARA-X, pySigma, and Microsoft's Kusto.Language analyzer on 22 November 2026. Test and tune them locally before production blocking.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. FBI Cyber Division / IC3. Threat Overview of ARTEX Autonomous Penetration Testing Framework (`FLASH-20261113-001` , 13 November 2026). https://www.ic3.gov/csa/2026/261113.pdf (verified on 22 November 2026)
2. BleepingComputer. FBI warns ARTEX AI hacking tool is powering global cyberattacks. https://www.bleepingcomputer.com/news/security/fbi-warns-artex-ai-hacking-tool-is-powering-global-cyberattacks/ (verified on 22 November 2026)
3. Censys ARC (Silas Cutler). ARTEX: Agentic Asbestos (5 November 2026). https://censys.com/blog/artex-agentic-asbestos/ (verified on 22 November 2026)
4. Microsoft Threat Intelligence. Storm-0940 uses ARTEX agentic AI (21 October 2026; framework context only). https://www.microsoft.com/en-us/security/blog/2026/10/21/storm-0940-uses-artex-agentic-ai-connectwise-supply-chain-compromise/ (verified on 22 November 2026)
5. AhnLab ASEC. Analysis of the infrastructure of ARTEX. https://asec.ahnlab.com/en/95875/ (verified on 22 November 2026)
6. Google Cloud. Best practices for managing service account keys. https://docs.cloud.google.com/iam/docs/best-practices-for-managing-service-account-keys (verified on 22 November 2026)
7. Google Cloud Storage. JSON API v1, Objects: insert. https://docs.cloud.google.com/storage/docs/json_api/v1/objects/insert (verified on 22 November 2026)
8. Google Identity. Using OAuth 2.0 for Server to Server Applications. https://developers.google.com/identity/protocols/oauth2/service-account (verified on 22 November 2026)
9. MITRE ATT&CK. T1595.002 Active Scanning: Vulnerability Scanning. https://attack.mitre.org/techniques/T1595/002/ (verified on 22 November 2026)
10. OWASP Cheat Sheet Series. Secrets Management Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html (verified on 22 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 (2) 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
R53-IOC-01	SHA-256	20745360f904db566a373ce39bb05ea52fee24c5f6a404db756e9c5aecb1981e		deploy_gcs_webshell.py · Operation LedgerSwarm follow-up playbook (iteration 1, 137 lines)	HUNT
R53-IOC-02	SHA-256	17c96f2ab239b6fd5b5454aae51301bee2021f0b8d9726c567c48b8ab167fbdc		deploy_gcs_webshell2.py · Operation LedgerSwarm follow-up playbook (iteration 2, 126 lines)	HUNT

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_ledgerswarm_artex_and_gcs_playbook_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Monograph 53 / 53 YARA Artifact Suite
   Operation LedgerSwarm: Structural rules for ARTEX console bundles,
   Google Cloud Storage-to-WordPress follow-up playbooks, and pax_cmd gates.
   Validation: Compiles cleanly in YARA-X; not replayed against live traffic.
   Classification: TLP:CLEAR
*/

rule ODIN_ARTEX_Console_and_Task_Bundle
{
    meta:
        author = "Odin by Brandefense"
        monograph = "53 / 53"
        classification = "TLP:CLEAR"
        description = "Matches ARTEX web console bundles combining title markers with task API routes or static
icon digest"
    strings:
        $title_en = "ARTEX - Autonomous Red-Team Expert" ascii wide
        $title_zh = { e6 94 bb e9 98 b2 e5 ae 89 e5 85 a8 e8 bf 90 e8 90 a5 }
        $fav_md5 = "cd3ebd16437d6d3963592f7365dd3980" ascii nocase
        $api_tasks = "/api/v1/tasks" ascii
        $dom_root = "artex-swarm-root" ascii
    condition:
        filesize < 10MB and (
            (($title_en or $title_zh) and ($api_tasks or $dom_root or $fav_md5)) or
            ($fav_md5 and ($api_tasks or $dom_root))
        )
}

rule ODIN_GCS_WordPress_PaxCmd_Deployer_Playbook
{
    meta:
        author = "Odin by Brandefense"
        monograph = "53 / 53"
        classification = "TLP:CLEAR"
        mitre_attack = "T1078.004"
        description = "Matches Python scripts exchanging Google service-account JWTs to overwrite WordPress theme
files"
    strings:
        $oa1 = "oauth2.googleapis.com/token" ascii
        $oa2 = "urn:ietf:params:oauth:grant-type:jwt-bearer" ascii
        $sc1 = "devstorage.full_control" ascii
        $up1 = "uploadType=media" ascii
        $wp1 = "wp-content/themes/" ascii
        $px1 = "pax_cmd" ascii
    condition:
        filesize < 128KB and ($oa1 or $oa2) and $sc1 and $up1 and ($wp1 or $px1)
}

rule ODIN_WordPress_PaxCmd_Injected_Theme_Gate
{
    meta:
        author = "Odin by Brandefense"
        monograph = "53 / 53"
        classification = "TLP:CLEAR"
        mitre_attack = "T1505.003"
        description = "Matches WordPress theme files containing the pax_cmd parameter gate and PHP execution
fallbacks"
    strings:
        $gate = "pax_cmd" ascii
```

```
        $req = "_REQUEST" ascii
        $fn1 = "passthru" ascii
        $fn2 = "shell_exec" ascii
        $fn3 = "system" ascii
    condition:
        filesize < 256KB and $gate and $req and 2 of ($fn*)
}
```

odin_ledgerswarm_swarm_and_cloud_key_abuse.rules

SURICATA

```
# Odin by Brandefense - Monograph 53 / 53 Suricata 7.x Detection Suite
# Operation LedgerSwarm: Autonomous Swarm Consoles, Secret Sweeps, and Cloud-to-CMS Follow-Up
# Validation: Loads cleanly in Suricata 7.0.12; not replayed against live traffic
# Classification: TLP:CLEAR

alert http any any -> $HOME_NET any (msg:"ODIN [R53] ARTEX Autonomous Red-Team Console HTTP Response Banner";
flow:established,to_client; http.stat_code; bsize:3; content:"200"; http.response_body; bsize:>32;
content:"ARTEX"; fast_pattern; pcre:"/(Autonomous Red-Team Expert|
\xe6\x94\xbb\xe9\x98\xb2\xe5\xae\x89\xe5\x85\xa8\xe8\xbf\x90\xe8\x90\xa5|artex-swarm-root)/"; classtype:trojan-
activity; sid:9553001; rev:1; metadata:created_at 2026_11_22, deployment perimeter;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R53] ARTEX Multi-Worker Swarm Task Dispatch POST";
flow:established,to_server; http.method; bsize:4; content:"POST"; http.uri; bsize:>12; content:"/api/v1/tasks";
fast_pattern; http.request_body; bsize:>24; content:"target_scope"; classtype:attempted-recon; sid:9553002; rev:
1; metadata:created_at 2026_11_22, deployment egress_proxy, mitre_attack T1595.002;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R53] High-Rate Dotenv or Git Config Secret Sweep";
flow:established,to_server; http.method; bsize:3; content:"GET"; http.uri; bsize:>4; content:"/."; fast_pattern;
pcre:"/\/\.(env(\.[a-z0-9_-]+)?|git\/config)(\?|$)/i"; threshold:type both, track by_src, count 5, seconds 60;
classtype:web-application-attack; sid:9553003; rev:1; metadata:created_at 2026_11_22, deployment
perimeter_and_waf, mitre_attack T1595.002;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R53] HUNT Non-Browser OAuth2 jwt-bearer Service-Account
Token Mint"; flow:established,to_server; http.method; bsize:4; content:"POST"; http.host; bsize:21;
content:"oauth2.googleapis.com"; http.request_body; bsize:>48; content:"urn%3Aietf%3Aparams%3Aoauth%3Agrant-
type%3Ajwt-bearer"; fast_pattern; content:"assertion="; classtype:policy-violation; sid:9553004; rev:1;
metadata:created_at 2026_11_22, deployment tls_egress_proxy, mitre_attack T1528;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R53] Google Cloud Storage uploadType=media PHP Write to
WordPress Path"; flow:established,to_server; http.method; bsize:4; content:"POST"; http.uri; bsize:>32;
content:"/upload/storage/v1/b/"; content:"uploadType=media"; fast_pattern; content:"wp-content";
http.content_type; bsize:17; content:"application/x-php"; classtype:web-application-attack; sid:9553005; rev:1;
metadata:created_at 2026_11_22, deployment tls_egress_proxy, mitre_attack T1505.003;)

alert http any any -> $HOME_NET any (msg:"ODIN [R53] WordPress Theme Backdoor pax_cmd Query Parameter Gate";
flow:established,to_server; http.uri; bsize:>8; content:"pax_cmd="; fast_pattern; classtype:web-application-
attack; sid:9553006; rev:1; metadata:created_at 2026_11_22, deployment perimeter_and_waf, mitre_attack
T1505.003;)

alert http any any -> $HOME_NET any (msg:"ODIN [R53] WordPress Theme Backdoor pax_cmd HTTP POST Body Gate";
flow:established,to_server; http.method; bsize:4; content:"POST"; http.request_body; bsize:>8;
content:"pax_cmd="; fast_pattern; classtype:web-application-attack; sid:9553007; rev:1; metadata:created_at
2026_11_22, deployment perimeter_and_waf, mitre_attack T1505.003;)

alert http any any -> $HOME_NET any (msg:"ODIN [R53] Direct HTTP Request for WordPress Theme 404.php or
functions.php"; flow:established,to_server; http.uri; bsize:>24; content:"/wp-content/themes/"; fast_pattern;
pcre:"/\/wp-content\/themes\/[^\/]+\/(404|functions)\.php/i"; classtype:web-application-attack; sid:9553008; rev:
1; metadata:created_at 2026_11_22, deployment perimeter_and_waf, mitre_attack T1505.003;)
```

sentinel_ledgerswarm_burst_and_cloud_key_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 53 / 53 Microsoft Sentinel & Defender XDR Hunt
// Operation LedgerSwarm: Correlates high-rate web URI scan bursts, secret-file probes,
// non-CI Google OAuth2/Storage API calls, and WordPress pax_cmd child shell execution.
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedCicdHosts = dynamic(["CMS-CICD-RUNNER-01", "GCP-TERRAFORM-RUNNER-01"]);
let ApprovedSecurityHosts = dynamic(["SEC-VALIDATION-LNX-01", "SEC-VALIDATION-WEB-01"]);
let CloudAuthAndStorageDomains = dynamic(["oauth2.googleapis.com", "storage.googleapis.com"]);
let WebWorkerBinaries = dynamic(["php-fpm", "php-cgi", "php", "apache2", "httpd", "lsphp", "w3wp.exe"]);
let SuspiciousChildBinaries = dynamic(["id", "whoami", "uname", "sh", "bash", "dash", "curl", "wget",
"python3"]);
union isfuzzy=true
(
    W3CIISLog
    | where TimeGenerated >= ago(Lookback)
    | summarize
        DistinctURIs = dcount(csUriStem),
        TotalRequests = count(),
        ObservedUserAgents = make_set(csUserAgent, 5),
        SampleURIs = make_set(csUriStem, 8),
        ScannerURIs = dcountif(csUriStem, csUserAgent has_any ("nuclei", "sqlmap", "gobuster", "ffuf",
"subfinder", "katana", "ARTEX")),
        SecretURIs = dcountif(csUriStem, csUriStem startswith "/.env" or csUriStem startswith "/.git/config" or
csUriStem startswith "/graphql"),
        PaxProbeCount = countif(csUriQuery contains "pax_cmd=")
        by bin(TimeGenerated, 5m), Computer, cIP
    | where DistinctURIs >= 25 or ScannerURIs >= 10 or SecretURIs >= 8 or PaxProbeCount > 0
    | extend
        Timestamp = TimeGenerated,
        DeviceName = Computer,
        AlertDisposition = iff(PaxProbeCount > 0, "ISOLATE_PAX_CMD_WEB_GATE_PROBE", "HUNT_SWARM_URI_SCAN_BURST"),
        InitiatingProcessFileName = cIP,
        ArtifactDetail = strcat("uris=", tostring(DistinctURIs), " reqs=", tostring(TotalRequests), " sample=",
tostring(SampleURIs))
    | project Timestamp, DeviceName, AlertDisposition, InitiatingProcessFileName, ArtifactDetail
),
(
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedCicdHosts)) and not(DeviceName in~ (ApprovedSecurityHosts))
    | where (InitiatingProcessFileName in~ (WebWorkerBinaries) and FileName in~ (SuspiciousChildBinaries))
        or ProcessCommandLine has_any ("deploy_gcs_webshell", "pax_cmd=", "uploadType=media")
    | extend
        AlertDisposition = iff(InitiatingProcessFileName in~ (WebWorkerBinaries),
"ISOLATE_CMS_WORKER_SHELL_SPAWN", "ISOLATE_CLOUD_KEY_PLAYBOOK_CLI"),
        ArtifactDetail = strcat("child=", FileName, " cmd=", ProcessCommandLine)
    | project Timestamp, DeviceName, AlertDisposition, InitiatingProcessFileName, ArtifactDetail
),
(
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedCicdHosts)) and not(DeviceName in~ (ApprovedSecurityHosts))
    | where (RemoteUrl has_any (CloudAuthAndStorageDomains) and InitiatingProcessFileName in~ ("python",
"python3", "curl", "php", "php-fpm"))
        or RemoteUrl has "pax_cmd="
    | summarize
        FirstSeen = min(Timestamp),
        DistinctRemoteUrls = dcount(RemoteUrl),
        SampleRemoteUrls = make_set(RemoteUrl, 8),
        ConnectionCount = count()
        by DeviceName, InitiatingProcessFileName, bin(Timestamp, 15m)
    | extend
```

```
        AlertDisposition = "HUNT_NON_CI_CLOUD_STORAGE_OR_OAUTH_CALL",
        ArtifactDetail = strcat("conns=", tostring(ConnectionCount), " urls=", tostring(SampleRemoteUrls))
    | project Timestamp = FirstSeen, DeviceName, AlertDisposition, InitiatingProcessFileName, ArtifactDetail
)
| sort by Timestamp desc
```

sigma_ledgerswarm_swarm_and_gcs_theme_abuse.yml

SIGMA

```
title: Google Cloud Storage PHP Theme Overwrite Outside Approved CI/CD Identities
id: 6d19a4f0-8b23-4c71-9e14-202611225301
status: experimental
description: Detects Google Cloud Storage object creation or update calls writing PHP files or WordPress theme
templates (404.php, functions.php) by principals outside named deployment service accounts.
author: Odin by Brandefense
date: 2026/11/22
tags:
  - attack.defense-evasion
  - attack.t1078.004
  - attack.persistence
  - attack.t1505.003
logsource:
  product: gcp
  service: gcp.audit
detection:
  selection_method:
    gcp.audit.method_name:
      - 'storage.objects.create'
      - 'storage.objects.update'
  selection_cms_php_object:
    gcp.audit.resource_name|re: '(wp-content/(themes|plugins)/.*\.php|/404\.php$|/functions\.php$|/
pax_cmd\.php$)'
  filter_cicd:
    gcp.audit.authentication_info.principal_email:
      - 'wp-cicd-deployer@prod-cms-sync.iam.gserviceaccount.com'
      - 'terraform-media-sync@prod-cms-sync.iam.gserviceaccount.com'
  condition: selection_method and selection_cms_php_object and not filter_cicd
falsepositives:
  - Authorized CMS release pipelines syncing pre-built WordPress theme code into object storage buckets
level: critical
---
title: Container or Subagent Runtime Spawning Batch Reconnaissance Utilities
id: 4b72e8c1-3a19-4f50-8d62-202611225302
status: experimental
description: Identifies container runtimes spawning automated reconnaissance or exploitation binaries with non-
interactive batch flags characteristic of autonomous multi-worker swarms.
author: Odin by Brandefense
date: 2026/11/22
tags:
  - attack.execution
  - attack.t1609
  - attack.reconnaissance
  - attack.t1595.002
logsource:
  category: process_creation
  product: linux
detection:
  selection_parent:
    ParentImage|endswith:
      - '/containerd-shim-runc-v2'
      - '/containerd-shim'
      - '/dockerd'
  selection_tool:
    Image|endswith:
      - '/nuclei'
      - '/sqlmap'
      - '/subfinder'
      - '/httpx'
      - '/naabu'
```

```
      - '/ffuf'
      - '/gobuster'
  selection_flags:
    CommandLine|contains:
      - '-jsonl'
      - '--batch'
      - '-silent'
      - '-rate-limit'
  filter_validation_host:
    ComputerName:
      - 'SEC-VALIDATION-LNX-01'
  condition: selection_parent and selection_tool and selection_flags and not filter_validation_host
falsepositives:
  - Registered security validation runner hosts executing scheduled external attack surface scans
level: high
```

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/operation-ledgerswarm-one-operator-s-ai-agent-campaign-against-payments/
Short Read · brandefense.io/odin-research/blog/operation-ledgerswarm-inside-one-ai-campaign-on-payments-and-web3/
Defense Kit · brandefense.io/odin-research/defense-kit/