FULL REPORT · OPERATIONS · MONOGRAPH 57 / 58 · 19 MIN READ

Operation SapperSync: Agent Skill Staging, ImageMagick Risks, and Container Persistence

Inside a TCP 9099 workbench staging local-upload AI agent skill packages, ImageMagick MSL scripts, layered Alpine Linux archives, and an ownCloud 10.13.0.4 diagnostic capture

BY
Odin by Brandefense

PUBLISHED
19 November 2026

READING TIME
19 min read

EDITION
Version 1.1 · TLP:CLEAR

SCOPE
Prepared November 2026

1. Executive Summary

IN SHORT

Operation SapperSync ran on an East Asian cloud server on TCP 9099 recorded on 13–20 November 2026. Odin by Brandefense analyzed 400 deduplicated entries totaling 435.30 MB across 199 sized files. The workbench holds staged AI skill packs, 11 ImageMagick MSL scripts, eight Alpine layers, and one 6 November ownCloud 10.13.0.4 diagnostic report. Block MSL coders in policy.xml, update ownCloud Graph API to 0.2.1 or 0.3.1, sign agent skills, and enforce read-only containers.

Enterprise stacks now place AI agent skill loaders beside image converters and containerized file shares. Each service accepts user archives or media uploads inside Linux containers. Weak upload rules, open ImageMagick policies, or writable container roots give attackers room to test several paths at once.

Between 13 and 20 November 2026, Odin by Brandefense examined an open HTTP directory on TCP port 9099. That directory sat on an East Asian virtual private server. On 20 November 2026, the listing reported 399 active files and 23 directories.

Across captures from 13 to 20 November 2026, deduplication yields 400 unique entries. Those entries include 199 sized files totaling 435,295,482 bytes (435.30 MB) and 201 unsized entries. Hashing identifies 190 unique SHA-256 digests across the sized files.

The server holds a diagnostic report from an external ownCloud host and logs third-party scanner hits. Odin by Brandefense publishes zero IPv4 addresses and zero workbench SHA-256 digests. The server

address, scanner addresses, diagnostic filename, transfer filenames, and container hostname are withheld.

1. **Local-upload AI agent skill archives sit beside a 54-skill catalogue slice.**
The root folder holds a 918,252-byte full skills JSON index and a 251,977-byte slice with 54 skills. Two identical test ZIP files unpack to 290 bytes each and mark their origin as a local upload. A 5,304-byte shell script prepares the Sapper environment. These staged files show preparation, not confirmed use against an external AI platform.

2. **Eleven ImageMagick MSL scripts pair with PNG and PHP upload tests.**
Seven sized MSL scripts total 128,802 bytes alongside four unsized MSL entries. The root folder also holds 15 PNG files across three content classes and 13 PHP-family files. These files align with known image-coder policy risks and PNG profile-chunk read flaws.

3. **Eight Alpine Linux tarballs model startup and cron persistence.**
A persistence folder holds 10 sized files totaling 3,735,731 bytes, led by eight numbered tarballs. Those layers add an Alpine BusyBox rootfs, version and hotfix markers, and a wrapper entrypoint. Two hidden scripts target PHP web cache paths and database variables on startup and every 15 minutes.

4. **One ownCloud 10.13.0.4 diagnostic report confirms external output without proving initial access.**
The root folder holds a 349,235-byte diagnostic file dated 6 November 2026 from containerized ownCloud 10.13.0.4 Community. It also holds a 1,986-line HTTP hit log. Older ownCloud builds carried vulnerable graphapi vendor files, yet this report alone does not show how access occurred.

Three Decisions for Security Leaders

1. **Require signed AI agent skill packages from internal repositories.** Disable ad-hoc local ZIP uploads on production agent gateways. Verify skill signatures in CI and run agent tools in isolated sandboxes.
2. **Lock down ImageMagick policy files and patch ownCloud appliances.** Block MSL, MVG, TEXT, HTTPS, and indirect coders in ImageMagick policy files and update past PNG profile flaws. Update ownCloud Graph API to 0.2.1 or 0.3.1 (bundled from 10.13.1), delete GetPhpInfo.php, and rotate container secrets.
3. **Enforce read-only container root filesystems and immutable digests.** Mount container root filesystems read-only so web workers cannot write startup scripts or cron jobs. Pin images by SHA-256 digest and monitor hidden binary paths.

2. Workbench Architecture, 400-Entry Inventory, and Operational Context

Operation SapperSync takes its name from two staged files on the `TCP 9099` East Asian cloud server (address withheld). Those files are `/deploy_sapper_v2.sh` (**5,304** bytes) in root and `usr/local/bin/.agent_sync` (**1,471** bytes unpacked) inside `/la/L6.tgz`. Defenders can use cyber threat intelligence to track how operators pair new AI skill tests with container persistence.

On `20 November 2026`, the open HTTP directory on `TCP 9099` reported **399 active files and 23 directories**. Across captures recorded on `13–20 November 2026`, deduplication yields **400 unique entries**. One transient entry from an earlier capture rotated out before the `20 November` listing.

Teams can also apply vulnerability intelligence when reviewing the ImageMagick and ownCloud version context tied to these files. Figure 1 maps the four workbench tracks and withholding rules.

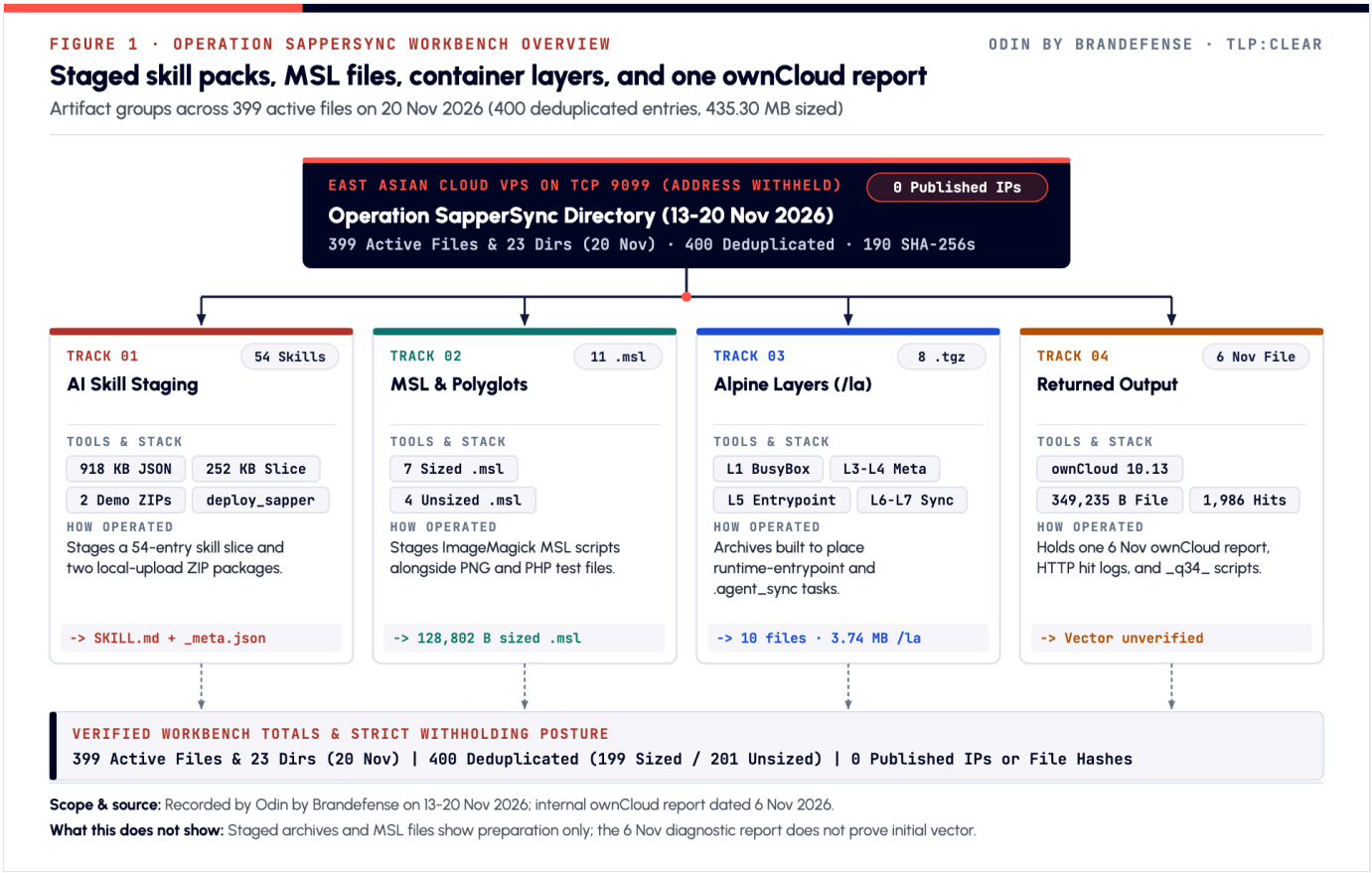


Figure 1: Staged skill packs, MSL files, container layers, and one ownCloud report

Workspace Zone Partition and File-Class Breakdown (`n = 400`)

Within the **400 deduplicated entries**, **199** sized files total **435,295,482 bytes (435.30 MB)**. The remaining **201** entries omit byte sizes in directory listings.

The **199** sized files span **190 unique SHA-256 digests**, all withheld. Nine sized files are exact byte duplicates, including the twin `demo-skill.zip` and `demo.zip` archives. Table 1 reconciles all **400** entries across the five top-level zones.

Workspace zone	Deduplicated entries	Share (%)	Sized files	Unsized entries	Sized bytes	Operational role on TCP 9099
Root Workspace `(root)`	387	96.75%	189	198	431,559,751	Staging scripts, **11** `.msl` files, **54**-skill slice, **2** skill ZIPs, `6 Nov` ownCloud report (**349,235** B), and **1,986** hit lines.
Alpine Layers `/la`	10	2.50%	10	0	3,735,731	Holds **8** numbered `.tgz` layers (`L1.tgz`–`L8.tgz`, **3,733,376** B) plus **2** sized companion files (**2,355** B).
Task Folder `/q33`	1	0.25%	0	1	0	Subfolder holding the unsized Python script `/q33/patch33.py`.
Output Folder `/q33o`	1	0.25%	0	1	0	Companion output subfolder holding **1** unsized entry.
Multiplexer `/tmux-0`	1	0.25%	0	1	0	Terminal multiplexer folder holding **1** unsized session socket entry.
Total deduplicated set	**400**	**100.00%**	**199**	**201**	**435,295,482**	**Matches 399 active files and 23 dirs on `20 Nov 2026` (190 SHA-256s withheld).**

Most workbench activity sits directly in root. The root zone holds **96.75%** of entries (**387** of **400**) and **99.14%** of sized bytes (**431,559,751** bytes). Only `/la` holds sized subdirectory files (**10** files, **3,735,731** bytes).

Table 2 reconciles the exact same **400** deduplicated entries (**199** sized files, **201** unsized entries, **435,295,482** bytes) across six mutually exclusive content classes.

Artifact class	Deduplicated entries	Share (%)	Sized files	Unsized entries	Sized bytes	Representative extensions & contents
Text, HTML & MSL (`text`)	187	46.75 %	106	81	7,026,344	Led by 74 `.html`, 30 extensionless, 19 `.txt`, 11 `.json`, 10 `.log`, 10 `.out`, 8 `.png`, 7 sized `.msl`, 5 `.jar`, and 4 `.b64` files.
Operator Scripts (`script`)	157	39.25 %	75	82	5,103,159	Exact split: 82 `.py`, 57 `.sh`, 10 `.php`, 6 `.js`, 1 `.png`, and 1 `.phtml`.
Unsized & Polyglots (`other`)	31	7.75%	0	31	0	Exact split: 15 extensionless, 4 unsized `.msl`, 3 `.b64`, 3 `.out`, 1 `.sock`, 1 `.lock`, 1 `.hdr`, 1 `.533805`, 1 `.php5`, and 1 `.phar`.
Archives (`archive`)	14	3.50%	12	2	423,164,869	Exact split: 9 `.tgz` (including `/la/L1.tgz` – `L8.tgz`), 3 `.gz`, and 2 `.zip` (`demo-skill.zip`, `demo.zip`); 97.21% of sized bytes.
Image Fixtures (`image`)	7	1.75 %	2	5	74	Exact split: 6 `.png` and 1 `.gif` small image test files.
Keys & Secrets (`secret`)	4	1.00%	4	0	1,036	Exact split: 2 extensionless key files, 1 `.openssh` key, and 1 `.pem` key (withheld).
Total deduplicated set	**400**	**100.00 %**	**199**	**201**	**435,295,482**	**Exact byte-level parity with Table 1** (435.30 MB).

Figure 5 shows how entry counts and byte volume split across these six classes. Text outputs (**187**) and scripts (**157**) make up **86.00%** (**344** of **400**) of entries. Archives (**14** entries, **12** sized) hold **97.21%** (**423,164,869** bytes) of sized bytes.

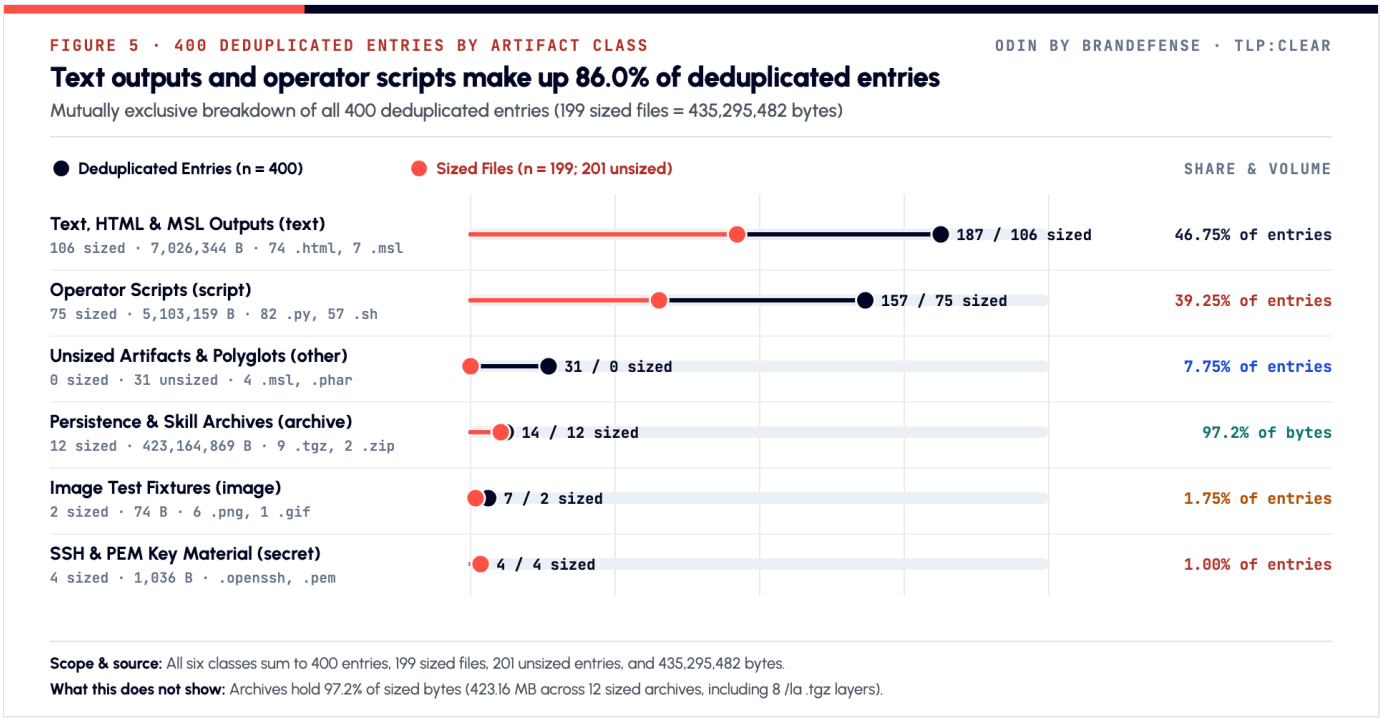


Figure 5: Text outputs and operator scripts make up 86.0% of deduplicated entries

Iterative Script Waves (`_q34_*` and `ch*.py`) and the 1,986-Line Hit Log

File names and sizes across the **157** scripts (**82** `.py` and **57** `.sh`) show an iterative testing workflow rather than one commercial tool. Table 3 summarizes the script groups and root access log.

Script or log group	Representative files	Recorded byte range	Role on workbench
Sapper & `_q34_*` shell set	`/deploy_sapper_v2.sh`, `/_q34_recon.sh`, `/bruteforce.sh`, `/_q34_s2.sh` – `s38.sh`	687–5,304 B	Environment setup and numbered target recon tasks.
`_q34_*` Python analyzers	`/analyze_34.py`, `/_q34_diff.py`, `/_q34_rhythm.py`, `/_q34_mem2.py`, `/_q34_mem.py`	528–1,981 B	Response diffing, timing analysis, and memory checks.
`ch*.py` & helper utilities	`/ch.py` – `/ch11.py`, `/count_loki.py`, `/dbg.py`, `/q33/patch33.py`	201–2,005 B (plus unsized)	Channel tests, log counting, and state debugging.

Script or log group	Representative files	Recorded byte range	Role on workbench
Root HTTP hit log	Root HTTP access log (1,986 lines)	1,986 lines	Records inbound hits on `TCP 9099`; scanner IPs withheld.

3. AI Agent Skill Staging and ImageMagick MSL Upload Artifacts

Two staged artifact sets in root target web upload and agent ingestion workflows (`[REF-06]`, `[REF-08]`). One set prepares **local-upload AI agent skill archives**. The second set prepares **ImageMagick `.msl`, PNG, and PHP polyglot files**.

Figure 3 models how these staged sets (dashed arrows) sit beside the verified captures on `TCP 9099` (solid arrows).



Figure 3: Analytical model of staged skill, MSL, and container files with one output

AI Agent Skill Catalogue Slices (54 Skills) and `local-upload` ZIPs (290 B Unpacked)

Many AI agent platforms load skills from portable ZIP packages. Each package usually pairs a `SKILL.md` instruction file with a `_meta.json` manifest and optional scripts. When an agent gateway allows unsigned local uploads (`source=local-upload`), an untrusted ZIP can alter agent prompts or tool calls (`[REF-06]` , `[REF-09]`).

Four artifacts define the SapperSync skill-staging track:

- **Full skills catalogue (918,252 bytes).** A root JSON file stores **918,252** bytes of skill registry definitions. It provides a reference for manifest fields and package layout.
- **Filtered 54-skill slice (251,977 bytes).** A second JSON file keeps **251,977** bytes covering **54** selected skill entries.
- **Twin `local-upload` archives (`demo-skill.zip` and `demo.zip`).** Both `.zip` files in the `archive` class unpack to **290** bytes. They hold identical test content pairing `SKILL.md` with `_meta.json` (`source=local-upload`).
- **Deployment script (`/deploy_sapper_v2.sh` , 5,304 bytes).** This root shell script handles staging setup. No returned logs on the server show external ingestion of `demo-skill.zip` or `demo.zip` , so this set remains staged preparation.

Eleven ImageMagick `.msl` Scripts (128,802 Sized Bytes), 15 PNGs, and PHP Variants

Web portals and file-sharing apps often call **ImageMagick** (`magick` or `convert`) to resize avatars and build previews. Unhardened ImageMagick deployments create two well-known upload risks (`[REF-01]` , `[REF-02]` , `[REF-03]`):

1. **Enabled script coders (`CVE-2016-3714` and `MSL` , `[REF-02]` , `[REF-03]`).** ImageMagick **6.x** before `6.9.3-10` and **7.x** before `7.0.1-1` allowed unsafe delegate calls (CVE-2016-3714). Newer builds still include **Magick Scripting Language (`MSL`)**, an XML coder supporting `<image>` , `<read>` , and `<write>` tags. Unless `policy.xml` blocks `MSL` , `MVG` , `TEXT` , and `HTTPS` (`[REF-03]`), `MSL` inputs can read local files or write files into web folders (`[REF-06]`).
2. **PNG profile-chunk file disclosure (`CVE-2022-44268` , `[REF-01]`).** In ImageMagick `7.1.0-49` , parsing a PNG with a crafted `profile` text chunk embeds local file bytes inside the resized output PNG. Upstream releases `7.1.0-52` and `6.9.12-67` fixed this flaw (`[REF-01]`).

Table 4 lists all **11 `.msl` entries** alongside the cross-class PNG and PHP-family files staged in root.

Staged file or group	Class	Count	Recorded size (bytes)	Structural role in upload testing
`/e_z4.msl`	`text` (sized)	1	64,064	Padded **64 KB** ImageMagick MSL XML script in root.
`/e_z6.msl`	`text` (sized)	1	64,062	Companion **64 KB** MSL variant two bytes shorter than `/e_z4.msl`.
`/d.msl`	`text` (sized)	1	181	Compact **181**-byte MSL directive test file.
`/z4.msl`, `/z2.msl`, `/a.msl`, `/b.msl`	`text` (sized)	4	495 (151+127+111+106)	Four small MSL XML templates testing `<read>` and `<write>` tags.
`/p.msl`, `/z1.msl`, `/z3.msl`, `/z6.msl`	`other` (unsized)	4	Unsized (0 sized B)	Four additional MSL files without recorded byte sizes.
Cross-class `.png` files	`text` (8), `image` (6), `script` (1)	15	In class totals	**15** `.png` files spanning text-bearing, minimal (**74** B), and script polyglot tests.
PHP-family extensions	`script` (11), `other` (2)	13	In class totals	**10** `.php`, **1** `.phtml`, **1** `.php5`, and **1** `.phar` testing upload extension filters (`[REF-06]`).

The seven sized `.msl` files sum to **128,802 bytes**, and four unsized `.msl` files bring the total to **11**. Splitting **15** `.png` files across `text` (**8**), `image` (**6**), and `script` (**1**) reflects polyglot testing. Valid PNG headers or chunks can wrap XML or script text to test naive MIME checks (`[REF-06]`, `[REF-10]`).

4. Layered Alpine Container Persistence and Returned ownCloud Output

Two other workbench components connect container-layer design with external appliance output. The **`/la` folder** holds **10** sized files (**3,735,731** bytes), and root holds a **returned `ownCloud 10.13.0.4 Community` diagnostic report** (**349,235** bytes, dated `6 November 2026`). Figure 4 shows the `TCP 9099` workspace and `/la` archive stack.

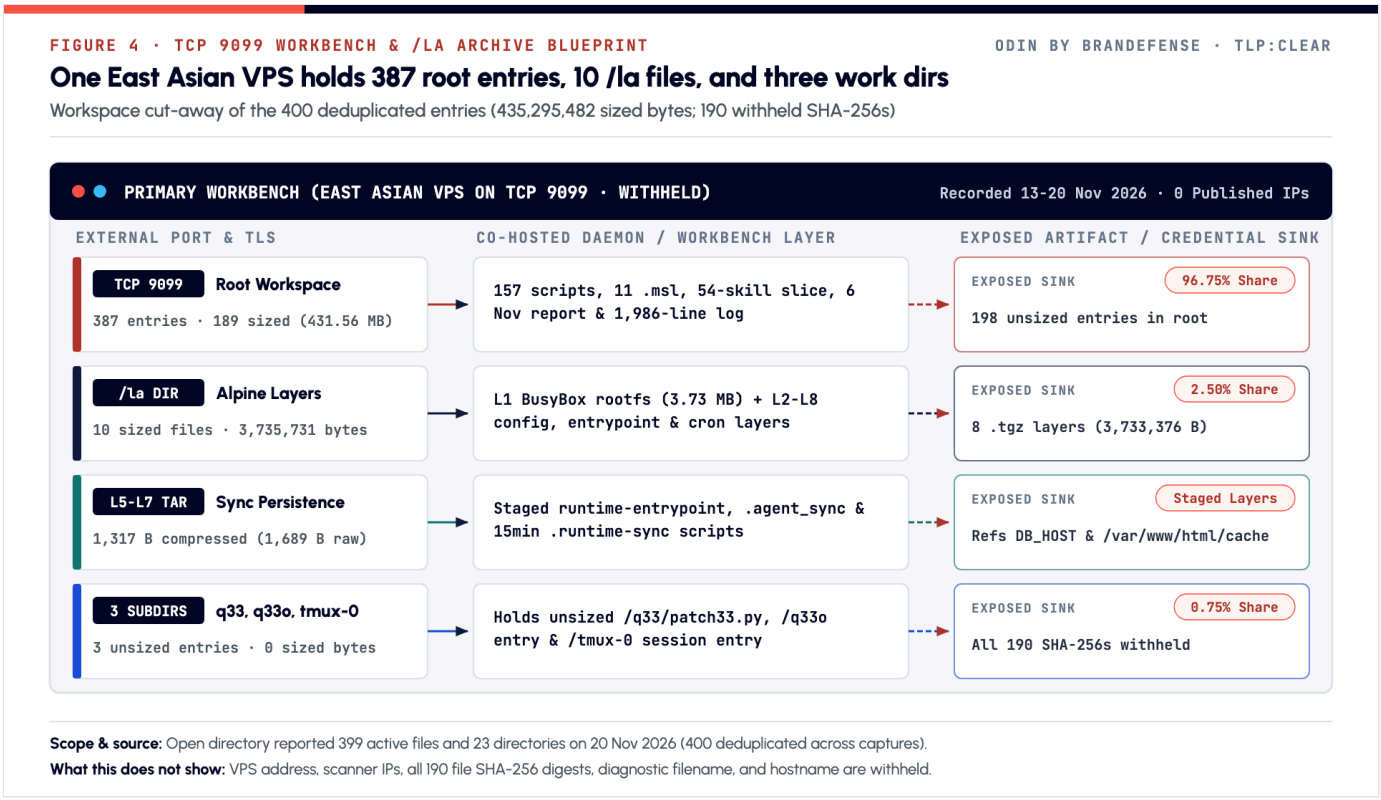


Figure 4: One East Asian VPS holds 387 root entries, 10 /la files, and three work dirs

Inside `/la/L1.tgz` Through `/la/L8.tgz` (3,733,376 Compressed Bytes)

Inside `/la`, eight numbered `.tgz` files (`L1.tgz` through `L8.tgz`) total **3,733,376** compressed bytes. Two small companion files (**2,355** bytes) complete the **10**-file, **3,735,731**-byte directory.

Each tarball mirrors an incremental OCI or Docker image layer (`[REF-07]`). Together they disguise hidden startup and cron scripts behind routine `release=2.4.2` and `channel=hotfix` markers (`[REF-09]`, `[REF-10]`). Table 5 describes all eight layers without printing shell code.

Archive path	Compressed (B)	Unpacked size / members	Target container path	Structural description
`/la/L1.tgz`	3,731,518	7,800,274 B (87 members)	`bin/busybox`, `etc/apk/*`	Alpine Linux BusyBox root filesystem layout (`bin/busybox` 808,712 B and `etc/apk` keys).
`/la/L2.tgz`	142	Compact layer	Base config path	Compact 142-byte configuration layer archive.

`/la/L3.tgz`	181	14 B (1 file)	`opt/runtime/VERSION`	Single-file tar writing `release=2.4.2` to `opt/runtime/VERSION`.
`/la/L4.tgz`	184	15 B (1 file)	`opt/runtime/CHANNEL`	Single-file tar writing `channel=hotfix` to `opt/runtime/CHANNEL`.
`/la/L5.tgz`	148	148 B (1 file)	`usr/local/bin/runtime-entrypoint`	POSIX shell entrypoint wrapper built to invoke `/usr/local/bin/.agent_sync` and sleep hourly.
`/la/L6.tgz`	912	1,471 B (1 file)	`usr/local/bin/.agent_sync`	Hidden script referencing `APP_ENV`, `DB_HOST`, `DB_PORT`, and `/var/www/html/cache`.
`/la/L7.tgz`	257	70 B (1 file)	`etc/periodic/15min/.runtime-sync`	BusyBox 15-minute cron task built to invoke `/usr/local/bin/.agent_sync`.
`/la/L8.tgz`	34	Minimal marker	Layer marker	Minimal 34-byte marker archive closing the eight-layer set.

Companion `/la` files (2)	**2,355**	Sized metadata	`/la/*`	Two remaining sized entries (`3,733,376 + 2,355 = 3,735,731` B across 10 files).

Three design choices in `L5.tgz` through `L7.tgz` stand out for container defenders (`[REF-07]`):

- **Redundant startup and 15-minute cron triggers.** `L5.tgz` places `usr/local/bin/runtime-entrypoint` to launch `.agent_sync` at container boot and loop hourly. `L7.tgz` places `etc/periodic/15min/.runtime-sync` so BusyBox `crond` re-runs `.agent_sync` every **15** minutes (`[REF-09]`).
- **Dot-prefixed hidden filenames (`[REF-10]`).** Both `.agent_sync` and `.runtime-sync` start with a leading dot. Standard `ls` checks without `-a` skip them, while `opt/runtime/VERSION` and `opt/runtime/CHANNEL` mimic a benign hotfix.
- **PHP web-container variable references.** The **1,471**-byte unpacked script in `L6.tgz` references `APP_ENV` , `DB_HOST` , `DB_PORT` , and `/var/www/html/cache` . Those names match containerized PHP apps that pass database settings in environment variables and mount `/var/www/html/cache` writable.

Returned `6 November 2026` `ownCloud 10.13.0.4 Community` Diagnostic File (349,235 Bytes)

The root workspace holds a **returned ownCloud diagnostic report (349,235 bytes; filename and hostname withheld)**. Its internal report date is `6 November 2026` , and it identifies `ownCloud 10.13.0.4 Community` on **containerized Linux**.

Defenders should separate verified version facts from unproven assumptions about initial entry:

- **Containerized ownCloud version context (`CVE-2023-49103` , `[REF-04]` , `[REF-05]`).** NIST NVD (`[REF-04]`) and ownCloud (`[REF-05]`) document `CVE-2023-49103` in `graphapi` **0.2.x** before `0.2.1` and **0.3.x** before `0.3.1` . The ownCloud 10.13.0 complete bundle included affected Graph API 0.3.0; the 10.13.1 complete bundle included fixed Graph API 0.3.1. Check separately installed app versions. On

containerized hosts, a vendor `GetPhpInfo.php` file under `/apps/graphapi/vendor/` exposed container environment secrets (`[REF-04]` , `[REF-05]`).

- **Initial-access vector remains unverified.** Although `10.13.0.4` belongs to the `10.13.0` line, holding a `10.13.0.4` diagnostic report does not establish how access occurred. Admin diagnostic exports can also follow stolen credentials, session cookies, or reverse-proxy flaws. Likewise, workbench files do not show whether `/la/L1.tgz` – `L8.tgz` were applied to that container.

Figure 2 links all five workbench artifact groups to the network, host, and file rules in Chapter 7.

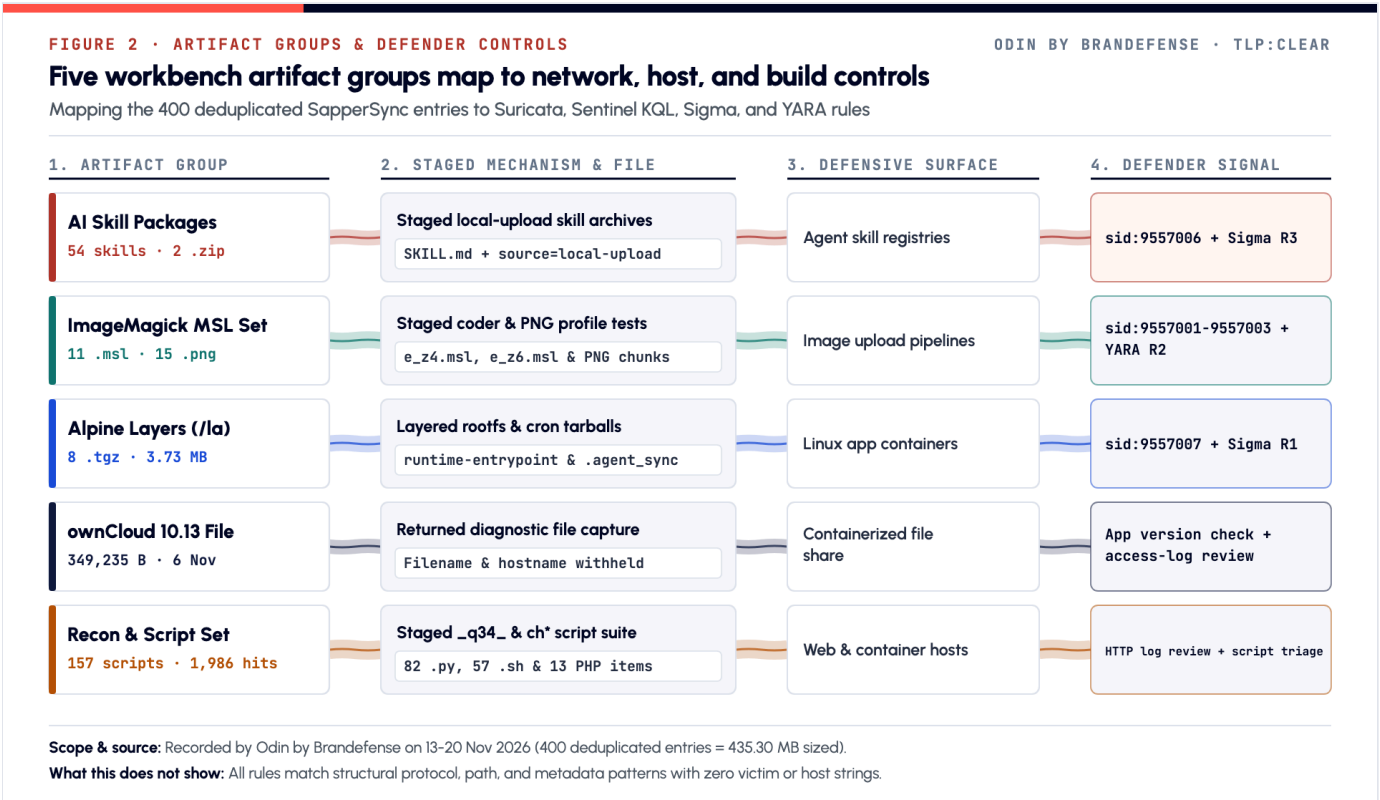


Figure 2: Five workbench artifact groups map to network, host, and build controls

5. Enterprise Upload, Skill Registry, and Container Exposure Analysis

The workbench highlights four defensive control areas. Teams should verify each control in their own environments before treating any gap as an established root cause:

Exposure layer	Observed SapperSync evidence	Control to verify	Operational risk
AI agent skill registries	918,252 B catalogue, 251,977 B slice (54 skills), `demo-skill.zip` & `demo.zip` (`source=local-upload`).	Verify whether agent gateways accept unsigned `local-upload` skill ZIPs or mutable `SKILL.md` files (`[REF-06]`).	Untrusted skill packs could steer agent prompts or invoke tools under service accounts.
ImageMagick upload pipelines	11 `.msl` files (128,802 B sized), 15 `.png` files, and 13 PHP-family files.	Verify ImageMagick version (`>= 7.1.0-52` / `6.9.12-67` , `[REF-01]`) and `policy.xml` blocks on `MSL` , `MVG` , and `HTTPS` (`[REF-03]`).	PNG `profile` chunk reads (`CVE-2022-44268`) or file read/write via `MSL` XML tags.
Containerized ownCloud hosts	Returned `6 Nov 2026` `ownCloud 10.13.0.4 Community` report (349,235 B; filename withheld).	Verify `graphapi >= 0.2.1` / `0.3.1` (`ownCloud >= 10.13.1` complete bundle or app update), removal of `GetPhpInfo.php` , and secret rotation (`[REF-05]`).	Container environment disclosure (`CVE-2023-49103`) or unauthorized admin config export.

Exposure layer	Observed SapperSync evidence	Control to verify	Operational risk
Container root mutability	`/la/L1.tgz` – `L8.tgz` (3,733,376 B) staging `runtime-entrypoint`, `.agent_sync`, and `15min/.runtime-sync`.	Verify `--read-only` container root filesystems, image digest pinning, and registry write access (`[REF-07]`).	Hidden startup wrappers or 15-minute cron tasks reading `DB_HOST` and `/var/www/html/cache`.

6. Alternative Hypotheses

Hypothesis	Evidence for	Evidence against	Assessment
`CVE-2023-49103` was the initial vector on the `ownCloud 10.13.0.4` host	Returned `6 Nov 2026` report (349,235 B) comes from containerized `ownCloud 10.13.0.4` (`10.13.0` bundle line, prior to `10.13.1` `graphapi 0.3.1` ; `[REF-04]` , `[REF-05]`).	No exploit request log for `/apps/graphapi/vendor/` appears on the server; admin exports also follow stolen logins or cookies.	**Unverified vector.** Cite `CVE-2023-49103` as version context only, and rotate secrets when patching.
Staged `/la/L1.tgz` – `L8.tgz` layers ran on the external target	`L6.tgz` references `APP_ENV` , `DB_HOST` , `DB_PORT` , and `/var/www/html/cache` , matching PHP containers.	Tarballs sit on `TCP 9099` without target container logs showing live extraction.	**Staged capability.** Hunt for `.agent_sync` and `.runtime-sync` without claiming live target deployment.

Hypothesis	Evidence for	Evidence against	Assessment
Benign developer or CTF test server	Holds `demo-skill.zip`, `ch*.py` scripts, and tiny `.msl` files (106–181 B).	Stores a 349,235-byte external `ownCloud 10.13.0.4` report, `/deploy_sapper_v2.sh`, `/bruteforce.sh`, hidden `.agent_sync` layers, and 4 key files.	**Contradicted.** Pairing external appliance output with hidden persistence layers reflects offensive staging.
All 1,986 HTTP hit lines come from targets	Root log records 1,986 inbound HTTP hits on `TCP 9099`.	Open non-standard cloud ports receive constant automated internet scanning.	**Mixed traffic.** Scanner IPv4s are third-party noise and are withheld (0 published IPs).

7. Validated Detection Rules

All server and scanner IPv4 addresses and all **190** workbench SHA-256 digests are withheld. Defenders should deploy **protocol, path, and structural file rules** that work across any staging host. Every rule passed syntax checks before release, including the Suricata linter (**0** problems), Sentinel KQL column validation, three Sigma rules, and four `filesize`-bounded YARA rules.

7.1 Suricata 7.x Network Signatures (`sid:9557001` -9557007)

These seven `alert` rules match multipart `.msl` uploads (`sid:9557001`), MSL `<image>` and `<write` tags (`sid:9557002`), and PNG `profilex00` keywords (`sid:9557003`). Rules `sid:9557004` and `sid:9557005` alert on ownCloud `graphapi` vendor PHP probes and `serverinfo` API calls. Rules `sid:9557006` and `sid:9557007` match `SKILL.md` `local-upload` posts and outbound `TCP 9099` `/la/L*.tgz` fetches.

All seven HTTP rules require cleartext traffic or an IDS tap behind a TLS-terminating reverse proxy or forward proxy. Because `alert` rules do not block inline, enforce active blocks in the WAF, proxy, or firewall.

```
SURICATA

# Odin by Brandefense - Monograph 57 (Operation SapperSync) Suricata 7.x Ruleset
# Validated against Odin Suricata linter (0 problems): covers MSL uploads, PNG profile chunks,
ownCloud diagnostic probes, local-upload skill metadata, and TCP 9099 layer fetches.

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound Multipart Imag
eMagick MSL File Upload"; flow:established,to_server; http.method; content:"POST"; bsize:4; ht
```

```
tp.request_body; content:"filename="; content:".msl|22|"; distance:0; nocase; fast_pattern; cl
asstype:web-application-attack; sid:9557001; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound ImageMagick MS
L XML Write Directive"; flow:established,to_server; http.method; content:"POST"; bsize:4; htt
p.request_body; content:"<image>"; nocase; content:"<write "; distance:0; nocase; fast_patter
n; classtype:web-application-attack; sid:9557002; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound PNG Raw Profil
e Metadata Upload"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.req
uest_body; content:"|89 50 4E 47 0D 0A 1A 0A|"; fast_pattern; content:"profile|00|"; distance:
0; classtype:web-application-attack; sid:9557003; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound ownCloud graph
api Vendor PHP Probe"; flow:established,to_server; http.method; content:"GET"; bsize:3; http.u
ri; content:"/apps/graphapi/vendor/"; nocase; fast_pattern; content:".php"; distance:0; nocas
e; classtype:web-application-attack; sid:9557004; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound ownCloud Serve
rinfo Diagnostic API Request"; flow:established,to_server; http.method; content:"GET"; bsize:
3; http.uri; content:"/ocs/v2.php/apps/serverinfo/api/v1/info"; nocase; fast_pattern; classtyp
e:attempted-recon; sid:9557005; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound Local-Upload A
I Agent Skill Package"; flow:established,to_server; http.method; content:"POST"; bsize:4; htt
p.request_body; content:"SKILL.md"; fast_pattern; content:"local-upload"; classtype:policy-vio
lation; sid:9557006; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET 9099 (msg:"ODIN-R57 SapperSync Outbound Alpine Conta
iner Layer Tarball Fetch on TCP 9099"; flow:established,to_server; http.method; content:"GET";
bsize:3; http.uri; content:"/la/L"; startswith; fast_pattern; content:".tgz"; distance:0; ends
with; classtype:trojan-activity; sid:9557007; rev:1;)
```

7.2 Microsoft Sentinel KQL Hunt

This query hunts hidden container persistence scripts, ImageMagick MSL coder invocations, local-upload skill writes, and selected process connections to `TCP 9099`. Replace the two placeholder hostnames in `ApprovedBuildHosts` with your own CI and media-worker host allowlist before scheduling it.

```kql
// Odin by Brandefense - Monograph 57 (Operation SapperSync)
// Hunts hidden container persistence scripts, ImageMagick MSL coder invocations, local-upload
skill writes, and selected process connections to TCP 9099.
let Lookback = 7d;
let ApprovedBuildHosts = dynamic(["app-ci-01.corp.example", "img-worker-01.corp.example"]);
let ContainerAndMslProcess =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBuildHosts))
    | where (
        ProcessCommandLine has_any ("/usr/local/bin/runtime-entrypoint", "/usr/local/bin/.agen
t_sync", "/etc/periodic/15min/.runtime-sync")
        or (FileName in~ ("magick", "convert", "identify") and (ProcessCommandLine has ".msl"
```

```
or ProcessCommandLine has "msl:"))
        or (FileName in~ ("unzip", "python", "python3", "tar") and ProcessCommandLine has_any
("SKILL.md", "demo-skill.zip") and ProcessCommandLine has "_meta.json")
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "ContainerSyncOrMslProcess", De
tail = ProcessCommandLine;
let ContainerAndSkillFiles =
    DeviceFileEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBuildHosts))
    | where (
        FolderPath in ("/usr/local/bin", "/etc/periodic/15min", "/opt/runtime")
        and FileName in ("runtime-entrypoint", ".agent_sync", ".runtime-sync", "VERSION", "CHA
NNEL")
    ) or (FileName == "_meta.json" and FolderPath has_any ("/tmp", "/var/www", "/uploads"))
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType =
"PersistenceOrSkillFileWrite", Detail = strcat(FolderPath, "/", FileName, " via ", InitiatingP
rocessFileName);
let Port9099ArchiveEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBuildHosts))
    | where RemotePort == 9099
    | where InitiatingProcessFileName in~ ("curl", "wget", "python", "python3", "busybox", "s
h", "ash")
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType =
"Port9099StagingEgress", Detail = strcat(InitiatingProcessFileName, " -> ", RemoteIP, ":", tos
tring(RemotePort));
union ContainerAndMslProcess, ContainerAndSkillFiles, Port9099ArchiveEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), Signa
ls = make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceName, AccountName
| order by LastSeen desc
```

7.3 Portable Sigma and YARA Rules

- **Sigma (`sigma_sappersync_container_persistence_msl_and_skills.yml`).** Rule 1 detects Linux command lines referencing `/usr/local/bin/runtime-entrypoint`, `/usr/local/bin/.agent_sync`, or `/etc/periodic/15min/.runtime-sync`. Rule 2 catches `/magick`, `/convert`, `/identify`, or `/mogrify` receiving `.msl` or `msl:` inputs. Rule 3 detects `/unzip`, `/tar`, `/python`, or `/python3` referencing `demo-skill.zip` or `SKILL.md` archive or manifest names in command lines; verify `_meta.json` and `source=local-upload` separately in file telemetry.
- **YARA (`odin_sappersync_skill_msl_and_container_layers.yar`).** Four `filesize`-bounded structural rules match `local-upload` skill manifests (**< 64KB**) and MSL XML `<read` / `<write` scripts (**< 256KB**). Rule 3 detects `.agent_sync` wrappers (**< 64KB**), and Rule 4 hunts individual extracted `release=2.4.2` or `channel=hotfix` marker contents (**< 64KB**; verify paths separately).

8. Behavioral and Protocol Hunt Markers (0 Published IPs or Workbench SHA-256s)

No IPv4 addresses or workbench SHA-256 digests appear in the indicator feeds because the server holds an external `ownCloud 10.13.0.4` diagnostic file and scanner logs. Defenders should hunt using the 10 structural protocol, path, and metadata markers below.

Marker category	Pattern / structural artifact	Layer	Defensive hunt & control
Skill upload metadata	`source=local-upload` (`SKILL.md` + `_meta.json`)	HTTP / Agent host	Block unsigned local-upload skill ZIPs; alert with `sid:9557006`, Sigma Rule 3, and YARA Rule 1.
ImageMagick MSL script	`.msl` (`<image>`, `<read`, `<write`)	HTTP / Media host	Block `MSL` in `policy.xml`; alert with `sid:9557001`-9557002, Sigma Rule 2, and YARA Rule 2.
PNG profile chunk	PNG header + `profilex00` keyword	HTTP ingress	Patch ImageMagick to `>= 7.1.0-52` / `6.9.12-67` (`[REF-01]`); alert on visible HTTP with `sid:9557003`.

Marker category	Pattern / structural artifact	Layer	Defensive hunt & control
Container entrypoint	`/usr/local/bin/runtime-entrypoint`	Linux container	Enforce read-only rootfs; hunt with Sentinel KQL, Sigma Rule 1, and YARA Rule 3.
Hidden sync script	`/usr/local/bin/.agent_sync` (`DB_HOST`, `/var/www/html/cache`)	Linux container	Inspect `/usr/local/bin` with `ls -la` or file integrity monitoring; alert via KQL and Sigma Rule 1.
BusyBox periodic cron	`/etc/periodic/15min/.runtime-sync`	Linux container	Disable unneeded container `crond` services and alert on `/etc/periodic/*` writes.
Layer hotfix markers	`/opt/runtime/VERSION` (`2.4.2`) & `/opt/runtime/CHANNEL` (`hotfix`)	Container image	Hunt individual extracted marker contents with YARA Rule 4 and verify `/opt/runtime/` paths separately via KQL.
ownCloud `graphapi` URI	`GET /apps/graphapi/vendor/*.php`	HTTPS / Proxy	Update `graphapi` to `>= 0.2.1` / `0.3.1` (`ownCloud >= 10.13.1` bundle, `[REF-05]`), delete `GetPhpInfo.php`, and alert via `sid:9557004`.

Marker category	Pattern / structural artifact	Layer	Defensive hunt & control
ownCloud diagnostic API	`GET /ocs/v2.php/apps/serverinfo/api/v1/info`	HTTPS / Proxy	Restrict admin diagnostic endpoints with MFA and IP allowlists (`sid:9557005`).
Layer staging URI	`GET /la/L*.tgz` on `TCP 9099`	HTTP / TCP egress	Block unexpected container egress to port 9099; alert on `/la/L*.tgz` fetches with `sid:9557007` and on selected process connections to `TCP 9099` with KQL.

9. Hardening Priorities for Enterprise Defenders

Platform, application, and container security teams should apply five practical controls in priority order:

1. **Patch containerized ownCloud and rotate secrets (`0-48 Hours`).** Update `graphapi` to `>= 0.2.1` or `>= 0.3.1` (included from the `ownCloud 10.13.1` complete bundle; check separately installed app versions). Delete `/apps/graphapi/vendor/microsoft/microsoft-graph/tests/GetPhpInfo.php` (`[REF-04]` , `[REF-05]`). Rotate database passwords, ownCloud admin passwords, S3 keys, and secret tokens stored in container environment variables (`[REF-05]`).
2. **Lock down ImageMagick `policy.xml` and update binaries (`0-72 Hours`).** Upgrade ImageMagick to `>= 7.1.0-52` or `>= 6.9.12-67` to close `CVE-2022-44268` (`[REF-01]`). Set `rights="none"` in `policy.xml` for `MSL` , `MVG` , `TEXT` , `EPHEMERAL` , `URL` , `HTTPS` , `HTTP` , and `@*` indirect reads (`[REF-02]` , `[REF-03]`). Re-encode user images inside isolated, networkless worker containers (`[REF-06]`).
3. **Disable unsigned `local-upload` AI agent skills (`Days 1-7`).** Turn off `source=local-upload` ZIP uploads on production AI agent gateways. Require CI-signed `SKILL.md` and `_meta.json` packages from internal repositories, and isolate agent tool runners from database secrets.
4. **Enforce read-only container root filesystems (`Days 1-7`).** Run web and file-sharing containers with `--read-only` (`readOnlyRootFilesystem: true` in Kubernetes, `[REF-07]`). Mount `tmpfs` with

`noexec,nosuid,nodev` only on required scratch paths so web workers cannot write `/usr/local/bin/.agent_sync` or `/etc/periodic/15min/.runtime-sync`. Pin container images by `@sha256:` digest.

5. **Block script execution on upload paths and filter egress (`Days 7-14`).** Disable PHP and script handlers on upload directories so `.php`, `.phtml`, `.php5`, and `.phar` files never run (`[REF-06]`). Restrict container egress to approved internal databases and APIs, blocking outbound connections to non-standard ports like `TCP 9099`.

10. Scope, Limits, and External References

Odin by Brandefense analyzed workbench files recorded between `13 and 20 November 2026` on `TCP 9099` of an East Asian cloud VPS. On `20 November 2026`, the open directory reported **399** active files and **23** directories. Deduplication across `13-20 November` captures yields **400** entries (**199** sized files totaling **435,295,482** bytes and **201** unsized entries, with **190** unique SHA-256 digests).

Staged `.msl` scripts, `local-upload` skill ZIPs, and `/la/L1.tgz` – `L8.tgz` tarballs show operator preparation rather than verified target execution. Likewise, the `6 November 2026` `ownCloud 10.13.0.4 Community` diagnostic report (**349,235** bytes) does not establish whether `CVE-2023-49103` or another method gave initial access. The VPS IPv4 address, scanner IPs in the **1,986**-line hit log, all **190** SHA-256 digests, diagnostic and cookie filenames, and the container hostname are withheld.

1. `[REF-01]` NIST National Vulnerability Database, *CVE-2022-44268 - ImageMagick PNG Profile Chunk Information Disclosure* (verified on 24 November 2026).
2. `[REF-02]` NIST National Vulnerability Database, *CVE-2016-3714 - ImageMagick Coder Policy Vulnerability* (verified on 24 November 2026).
3. `[REF-03]` ImageMagick Studio LLC, *ImageMagick Security Policy (`policy.xml` Hardening Guide)* (verified on 24 November 2026).
4. `[REF-04]` NIST National Vulnerability Database, *CVE-2023-49103 - ownCloud graphapi Containerized Information Disclosure* (verified on 24 November 2026).
5. `[REF-05]` ownCloud Security Advisories, *Disclosure of Sensitive Credentials and Configuration in Containerized Deployments* (verified on 24 November 2026).
6. `[REF-06]` OWASP Foundation, *Unrestricted File Upload Vulnerability Defense Guidance* (verified on 24 November 2026).
7. `[REF-07]` Docker Documentation, *Docker Engine Security and Container Hardening* (verified on 24 November 2026).
8. `[REF-08]` MITRE ATT&CK, *Exploit Public-Facing Application (T1190)* (verified on 24 November 2026).

9. [REF-09] MITRE ATT&CK, *Persistence Tactic (TA0003)* (verified on 24 November 2026).
10. [REF-10] MITRE ATT&CK, *Stealth Tactic (TA0005)* (verified on 24 November 2026).

BRANDEFENSE CAPABILITIES RELATED TO THIS RESEARCH

Cyber Threat Intelligence

Track the actors, infrastructure and campaigns behind threats like these.

DEFENDER KIT

Indicators of compromise

Verified adversary infrastructure and tool hashes from this study. Every value below is also discussed in the report.

No network or file indicators are published for this study

The hosts in this report are exposed or unattributed systems rather than confirmed adversary infrastructure, so they stay anonymous. Use the detection rules below to find the same exposure in your own estate.

10 records from this study are not published in this table: candidate, suspected and dual-use hosts, victim-related values, and hashes of exposed secrets or files.

DEFENDER KIT

Detection rules

The rule files exactly as printed in the report. Test them in your own engine before deployment.

Suricata `odin_sappersync_msl_skill_and_owncloud_hunt.rules` 11 lines

```
# Odin by Brandefense - Monograph 57 (Operation SapperSync) Suricata 7.x Ruleset
# Validated against Odin Suricata linter (0 problems): covers MSL uploads, PNG profile chunk
s, ownCloud diagnostic probes, local-upload skill metadata, and TCP 9099 layer fetches.

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound Multipart Ima
geMagick MSL File Upload"; flow:established,to_server; http.method; content:"POST"; bsize:4;
http.request_body; content:"filename="; content:".msl|22|"; distance:0; nocase; fast_pattern;
classtype:web-application-attack; sid:9557001; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound ImageMagick M
SL XML Write Directive"; flow:established,to_server; http.method; content:"POST"; bsize:4; ht
tp.request_body; content:"<image>"; nocase; content:"<write "; distance:0; nocase; fast_patte
rn; classtype:web-application-attack; sid:9557002; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound PNG Raw Profi
le Metadata Upload"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.r
equest_body; content:"|89 50 4E 47 0D 0A 1A 0A|"; fast_pattern; content:"profile|00|"; distan
ce:0; classtype:web-application-attack; sid:9557003; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound ownCloud grap
hapi Vendor PHP Probe"; flow:established,to_server; http.method; content:"GET"; bsize:3; htt
p.uri; content:"/apps/graphapi/vendor/"; nocase; fast_pattern; content:".php"; distance:0; no
case; classtype:web-application-attack; sid:9557004; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound ownCloud Serv
erinfo Diagnostic API Request"; flow:established,to_server; http.method; content:"GET"; bsiz
e:3; http.uri; content:"/ocs/v2.php/apps/serverinfo/api/v1/info"; nocase; fast_pattern; class
type:attempted-recon; sid:9557005; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R57 SapperSync Inbound Local-Upload
AI Agent Skill Package"; flow:established,to_server; http.method; content:"POST"; bsize:4; ht
tp.request_body; content:"SKILL.md"; fast_pattern; content:"local-upload"; classtype:policy-v
iolation; sid:9557006; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET 9099 (msg:"ODIN-R57 SapperSync Outbound Alpine Cont
ainer Layer Tarball Fetch on TCP 9099"; flow:established,to_server; http.method; content:"GE
T"; bsize:3; http.uri; content:"/la/L"; startswith; fast_pattern; content:".tgz"; distance:0;
endswith; classtype:trojan-activity; sid:9557007; rev:1;)
```

Show full rule

YARA `odin_sappersync_skill_msl_and_container_layers.yar` 70 lines

```
/*
   Odin by Brandefense - Monograph 57 (Operation SapperSync) YARA Suite
   4 filesize-bounded structural rules covering local-upload AI agent skill manifests,
```

```
   ImageMagick MSL XML scripts, hidden container .agent_sync wrappers, and layered runtime ma
rkers.
*/

rule ODIN_SapperSync_LocalUpload_Agent_Skill_Meta
{
    meta:
        author = "Odin by Brandefense"
        monograph = "57"
        description = "Detects local-upload AI agent skill manifests and archives pairing SKI
LL.md or _meta.json structure with source=local-upload"
        date = "2026-11-24"
    strings:
        $s1 = "local-upload" ascii
        $m1 = "SKILL.md" ascii
        $m2 = "\"source\"" ascii
        $m3 = "source=" ascii
    condition:
        filesize < 64KB and $s1 and ($m1 or $m2 or $m3)
}

rule ODIN_SapperSync_ImageMagick_MSL_Polyglot_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "57"
        description = "Detects staged ImageMagick Magick Scripting Language XML files pairing
image blocks with read and write file directives"
        date = "2026-11-24"
    strings:
        $x1 = "<image>" ascii nocase
        $x2 = "</image>" ascii nocase
        $r1 = "<read " ascii nocase
        $w1 = "<write " ascii nocase
        $f1 = "filename=" ascii nocase
    condition:
        filesize < 256KB and ($x1 or $x2) and $r1 and $w1 and $f1
}

rule ODIN_SapperSync_Container_Entrypoint_AgentSync
{
    meta:
        author = "Odin by Brandefense"
        monograph = "57"
        description = "Detects staged container entrypoint, periodic cron, and hidden .agent_
sync scripts targeting web container environments"
        date = "2026-11-24"
    strings:
        $a1 = "/usr/local/bin/.agent_sync" ascii
        $p1 = "runtime-entrypoint" ascii
        $p2 = ".runtime-sync" ascii
```

```
        $e1 = "DB_HOST" ascii
        $e2 = "/var/www/html/cache" ascii
    condition:
        filesize < 64KB and $a1 and ($p1 or $p2 or ($e1 and $e2))
}

rule ODIN_SapperSync_Runtime_Hotfix_Layer_Markers
{
    meta:
        author = "Odin by Brandefense"
        monograph = "57"
        description = "Detects individual extracted container marker file contents containing
release=2.4.2 or channel=hotfix (verify /opt/runtime/VERSION or /opt/runtime/CHANNEL path sep
arately)"
        date = "2026-11-24"
    strings:
        $v2 = "release=2.4.2" ascii
        $c2 = "channel=hotfix" ascii
    condition:
        filesize < 64KB and ($v2 or $c2)
}
```

Sentinel KQL **sentinel_sappersync_container_sync_and_msl_hunt.kql** 34 lines

```
// Odin by Brandefense - Monograph 57 (Operation SapperSync)
// Hunts hidden container persistence scripts, ImageMagick MSL coder invocations, local-uploa
d skill writes, and selected process connections to TCP 9099.
let Lookback = 7d;
let ApprovedBuildHosts = dynamic(["app-ci-01.corp.example", "img-worker-01.corp.example"]);
let ContainerAndMslProcess =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBuildHosts))
    | where (
        ProcessCommandLine has_any ("/usr/local/bin/runtime-entrypoint", "/usr/local/bin/.age
nt_sync", "/etc/periodic/15min/.runtime-sync")
        or (FileName in~ ("magick", "convert", "identify") and (ProcessCommandLine has ".msl"
or ProcessCommandLine has "msl:"))
        or (FileName in~ ("unzip", "python", "python3", "tar") and ProcessCommandLine has_any
("SKILL.md", "demo-skill.zip") and ProcessCommandLine has "_meta.json")
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "ContainerSyncOrMslProcess", D
etail = ProcessCommandLine;
let ContainerAndSkillFiles =
    DeviceFileEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBuildHosts))
    | where (
        FolderPath in ("/usr/local/bin", "/etc/periodic/15min", "/opt/runtime")
```

```
        and FileName in ("runtime-entrypoint", ".agent_sync", ".runtime-sync", "VERSION", "CH
ANNEL")
    ) or (FileName == "_meta.json" and FolderPath has_any ("/tmp", "/var/www", "/uploads"))
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType =
"PersistenceOrSkillFileWrite", Detail = strcat(FolderPath, "/", FileName, " via ", Initiating
ProcessFileName);
let Port9099ArchiveEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBuildHosts))
    | where RemotePort == 9099
    | where InitiatingProcessFileName in~ ("curl", "wget", "python", "python3", "busybox", "s
h", "ash")
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType =
"Port9099StagingEgress", Detail = strcat(InitiatingProcessFileName, " -> ", RemoteIP, ":", to
string(RemotePort));
union ContainerAndMslProcess, ContainerAndSkillFiles, Port9099ArchiveEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), Sign
als = make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceName, AccountName
| order by LastSeen desc
```

Sigma **sigma_sappersync_container_persistence_msl_and_skills.yml** 88 lines

```
title: SapperSync Hidden Container Entrypoint and Periodic Agent Sync
id: 8c57a1e2-4b31-4d89-9a14-57c30e91a101
status: experimental
description: Detects Linux container processes invoking the staged runtime-entrypoint wrappe
r, hidden .agent_sync script, or 15-minute BusyBox periodic .runtime-sync task observed in Op
eration SapperSync.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_paths:
    CommandLine|contains:
      - '/usr/local/bin/runtime-entrypoint'
      - '/usr/local/bin/.agent_sync'
      - '/etc/periodic/15min/.runtime-sync'
  filter_build_hosts:
    ComputerName:
      - 'app-ci-01.corp.example'
  condition: selection_paths and not filter_build_hosts
falsepositives:
  - Custom container build validation runners on designated CI hosts
level: critical
tags:
  - attack.persistence
```

```
  - attack.defense-evasion
---
title: SapperSync ImageMagick MSL Coder Invocation on Linux
id: 8c57a1e2-4b31-4d89-9a14-57c30e91a102
status: experimental
description: Detects ImageMagick CLI utilities processing Magick Scripting Language (.msl or
msl:) inputs when policy.xml fails to block script coders.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_img:
    Image|endswith:
      - '/magick'
      - '/convert'
      - '/identify'
      - '/mogrify'
  selection_msl:
    CommandLine|contains:
      - '.msl'
      - 'msl:'
  filter_media_hosts:
    ComputerName:
      - 'img-worker-01.corp.example'
  condition: selection_img and selection_msl and not filter_media_hosts
falsepositives:
  - Approved offline batch media-conversion servers explicitly configured to use MSL template
s
level: high
tags:
  - attack.initial-access
  - attack.defense-evasion
---
title: SapperSync Local-Upload AI Agent Skill Archive Unpacking
id: 8c57a1e2-4b31-4d89-9a14-57c30e91a103
status: experimental
description: Detects archive or Python utilities referencing AI agent skill archive or manife
st names (demo-skill.zip or SKILL.md) in command lines outside approved CI runners; verify _m
eta.json and source=local-upload separately in file telemetry.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_tool:
    Image|endswith:
      - '/unzip'
      - '/tar'
```

```
      - '/python'
      - '/python3'
  selection_skill:
    CommandLine|contains:
      - 'demo-skill.zip'
      - 'SKILL.md'
  filter_ci_hosts:
    ComputerName:
      - 'skill-ci-01.corp.example'
  condition: selection_tool and selection_skill and not filter_ci_hosts
falsepositives:
  - Developer workstations or CI pipelines building internal agent skill packages
level: medium
tags:
  - attack.initial-access
  - attack.persistence
```

