FULL REPORT • POST-EXPLOITATION • MONOGRAPH 73 / 72 • 22 MIN READ

# Linux Staging Workbench: Polkit, OpenSSH, and Bore Tunnels

Inside a 313-entry Netherlands workbench combining staged web API fuzzing, Polkit and Dirty COW privilege-escalation helpers, UID-0 OpenSSH persistence, and Rust bore TCP tunnels


BY

Odin by Brandefense



PUBLISHED

19 November 2026

READING TIME

22 min read

EDITION

Version 1.1 · TLP:CLEAR

SCOPE

Prepared November 2026


## 1. Executive Summary


IN SHORT

Between 13 and 26 November 2026, Odin by Brandefense analyzed Workbench A, a 313-entry adversary staging directory on TCP port 8000 of a Netherlands server totaling 3.85 megabytes. The workspace pairs web API fuzzing and SQL probes with Polkit and Dirty COW escalation helpers, UID-0 OpenSSH scripts, and Rust bore tunnels. Patch CISA KEV baselines, audit OpenSSH AllowUsers policies, and restrict server TCP egress.


Linux servers running messaging APIs and partner web portals rely on layered perimeter, kernel, daemon, and network controls. An application flaw on an unpatched Linux host can expose local privilege-escalation tools, secondary root accounts, and outbound TCP tunnels. `Workbench A` brings each of those operational layers into one open directory.

Between `13 and 26 November 2026`, Odin by Brandefense studied `Workbench A` on `TCP 8000` of a Netherlands server. The live directory was first recorded on `13 November 2026` and last verified on `26 November 2026` at `22:09 UTC`. Across every artifact group, `Workbench A` is assessed consistently as an adversary staging workbench.

Deduplicated indexing captured all **313** reported entries. Those entries span **311** sized files totaling **3,847,708** bytes (**3.85** megabytes) and **2** unsized root entries, with **4** snapshot files retained in full.

`Workbench A` scopes a regional messaging gateway and holds captured HTTP request templates. Odin by Brandefense therefore publishes zero IPv4 addresses, zero file SHA-256 digests, and zero organization or customer names. Two target-suffixed upload script names are also withheld.

### First, web API discovery, HTTP templates, and frontend scripts make up 232 of the 313 entries.

Those **232** items (**1,560,301** bytes across **230** sized files and **2** unsized entries) map endpoints and parameters of a regional messaging gateway. Key files include `api_doc.pdf` (**635,631** bytes), `ffuf.log` (**293,402** bytes), **12** `/burp_requests/` templates, and **7** `/jsdl/` JavaScript files.

### Second, thirty-three shell scripts stage numeric SQL probes, schema checks, and multipart upload tests.

Totaling **20,215** bytes across **33** sized files, this group tests numeric SQL parameters, database schemas, and PHP upload handlers. Directory files alone do not confirm database extraction or upload execution on the external target.

### Third, ten local escalation items sit beside nine UID-0 OpenSSH persistence scripts.

Ten escalation files (**1,150,699** bytes) pair Dirty COW binary `/dc` (**1,146,296** bytes) with Polkit `pwnkit.c` (**1,267** bytes) and eight shell wrappers. Nine SSH scripts (**3,229** bytes) stage a secondary UID-0 `toor` account, RSA keys, and `sshd_config` `AllowUsers` edits.

### Fourth, twenty-seven Rust bore items and two cleanup scripts prepare outbound TCP relays and log deletion.

Archive `/bore.tgz` (**1,097,869** bytes) pairs with **26** shell scripts (**12,707** bytes) exposing local ports over `TCP 80` and **443**. Two cleanup scripts (**2,688** bytes) target host logs, though all **313** workspace entries remain staged preparation rather than verified target impact.

## Three Decisions for Enterprise Security Leaders

1. **Verify CISA KEV patch levels across Linux Polkit and kernel packages.** Confirm every DMZ and web Linux server runs Polkit packages with January 2022 vendor patches, verify each kernel against its vendor's Dirty COW fixed package, and upgrade unsupported releases.

2. **Audit Linux UID-0 accounts and OpenSSH AllowUsers directives daily.** Alert on any non-`root` account carrying UID **0** in `/etc/passwd` and lock `/etc/ssh/sshd_config` `AllowUsers` and `PermitRootLogin` settings under file integrity monitoring.
3. **Block direct server TCP egress and stream Linux audit logs off-host.** Force application server traffic through authenticated proxies to block raw TCP `bore` tunnels on ports **80**, **443**, and **7835**, while forwarding `auth.log` and `auditd` events in real time.

## 2. Workbench Architecture, 313-Entry Reconciliation, and Timeline Scope

`Workbench A` ran on a **Netherlands commercial virtual private server** (address withheld) assessed consistently as an **adversary staging workbench**. The server exposed an open HTTP directory on `TCP 8000` containing **258** iterative offensive `.sh` scripts, recon logs, escalation helpers, and tunnel archives. Infrastructure teams use vulnerability intelligence to spot unpatched Linux Polkit and kernel packages across internet-facing fleets.

Odin by Brandefense first recorded `Workbench A` on `13 November 2026` and last verified the directory on `26 November 2026` **at** `22:09 UTC`. Detection teams also apply cyber threat intelligence to turn staged shell wrappers and tunnel patterns into network and host rules. Figure 1 maps the four operational tracks across all **313** entries.

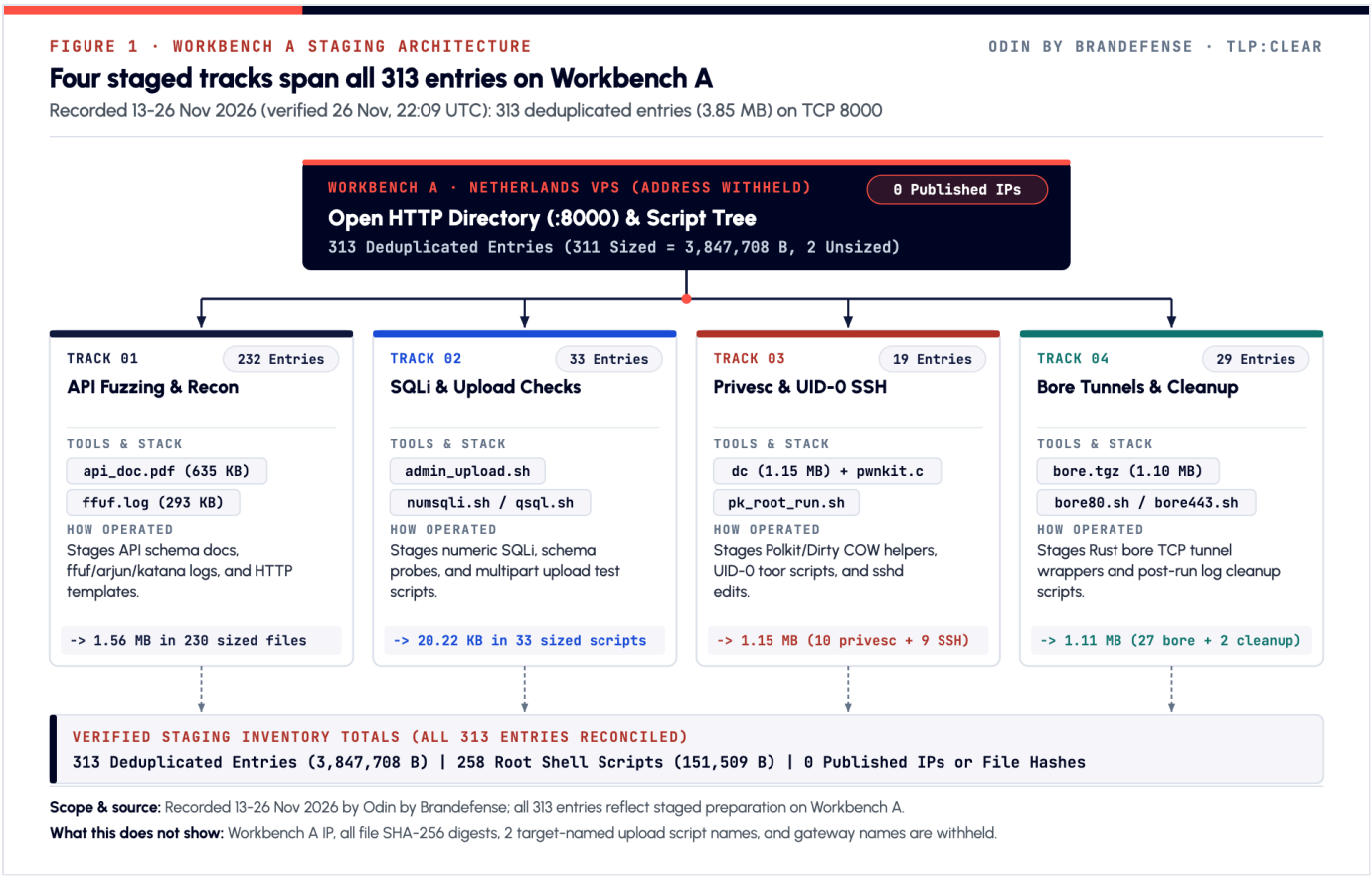


Figure 1: Four staged tracks span all 313 entries on Workbench A

## Complete 313-Entry Reconciliation Across Zones and File Types

`Workbench A`'s live directory listing reported **313 files**, matching our deduplicated count of **313 entries** one-for-one. Those **313** entries consist of **311 sized files** totaling **3,847,708 bytes (3.85 MB)** and **2 unsized extensionless entries** in `(root)`. Four snapshot files, including `ffuf.log` (**293,402** B), were stored in full.

Across folders, `(root)` holds **294 entries** (**292** sized files totaling **3,526,949** B plus **2** unsized items). Two subdirectories hold the remaining **19 sized entries** (**320,759** B). Folder `/burp_requests/` holds **12** HTTP templates (**2,461** B, built by `make_burp_requests.sh` [**918** B]; **3,379** B combined), and `/jsdl/` holds **7** JavaScript files (**318,298** B).

| Workspace zone | Subdirs | Deduplicated entries | Share (%) | Sized files | Unsized entries | Sized bytes | Operational role |
|---|---|---|---|---|---|---|---|
| **Root Iterative Shell Scripts (258 `.sh` Files)** | 0 (root) | 258 | 82.4% | 258 | 0 | 151,509 | 258 root `.sh` scripts covering API fuzzing, SQLi and upload checks, escalation wrappers, UID-0 `toor` edits, `bore` tunnels, and cleanup. |
| **Root Recon Logs, Wordlists & Binaries** | 0 (root) | 36 | 11.5% | 34 | 2 | 3,375,440 | `/dc` (1,146,296 B), `/bore.tgz` (1,097,869 B), `/api_doc.pdf` (635,631 B), `/ffuf.log` (293,402 B), recon logs, `pwnkit.c`, 1 `.php`, and 2 unsized items. |
| **Captured HTTP Templates (`/burp_requests/`)** | 1 | 12 | 3.8% | 12 | 0 | 2,461 | 12 HTTP templates (2,461 B; built by `make_burp_requests.sh` [918 B], 3,379 B combined) modeling messaging gateway calls. |
| **Frontend JavaScript Assets (`/jsdl/`)** | 1 | 7 | 2.2% | 7 | 0 | 318,298 | 7 frontend `.js` files (318,298 B), led by `app.js` (111,747 B), `jquery.js` (96,763 B), and `jquery.min.js` (94,840 B). |
| **Combined deduplicated inventory** | **2 subdirs + root** | **313** | **100.0%** | **311** | **2** | **3,847,708** | **Exact parity (294 root + 19 subdir entries; shares rounded to 0.1%).** |

A second breakdown reconciles the same **313** entries (**311** sized, **2** unsized, **3,847,708** bytes) by file kind. Scripts (**266** entries, **85.0%**) and text/log/source files (**42** entries, **13.4%**) represent **98.4%** of entry counts. Three root files ( `/dc` , `/bore.tgz` , and `/api_doc.pdf` ) account for **2,879,796** bytes (**74.8%** of sized bytes).

| Artifact class ( `kind` ) | Deduplicated entries | Share (%) | Sized files | Unsized entries | Sized bytes | Verified extensions & contents |
|---|---|---|---|---|---|---|
| **Executable & frontend scripts ( `script` )** | 266 | 85.0% | 266 | 0 | 470,163 | 258 root `.sh` scripts (151,509 B), 7 `.js` files in `/jsdl/` (318,298 B), and 1 root `.php` file (356 B). |
| **Logs, wordlists, templates & C source ( `text` )** | 42 | 13.4% | 42 | 0 | 497,749 | 31 `.txt` files ( `api_doc.txt` , `plist.txt` , `gau_*` , `/burp_requests/` ), 5 `.log` files ( `ffuf.log` , `arjun.log` , `httpx_install.log` , `katana.log` , 1 recon log), 4 `.json` , and 2 `.c` files. |
| **Unsized root items ( `other` )** | 2 | 0.6% | 0 | 2 | 0 | 2 extensionless unsized entries in `(root)` . |
| **Compiled Linux binaries ( `exe` )** | 1 | 0.3% | 1 | 0 | 1,146,296 | 1 extensionless binary `/dc` (1,146,296 B, 29.8% of sized bytes) paired with `cow_restart.sh` . |
| **Compressed tool archives ( `archive` )** | 1 | 0.3% | 1 | 0 | 1,097,869 | 1 `.tgz` archive `/bore.tgz` (1,097,869 B, 28.5% of sized bytes) staging the Rust `bore` binary. |
| **API specification documents ( `doc` )** | 1 | 0.3% | 1 | 0 | 635,631 | 1 `.pdf` document `/api_doc.pdf` (635,631 B, 16.5% of sized bytes) for a regional messaging gateway. |

| Artifact class (`kind`) | Deduplicated entries | Share (%) | Sized files | Unsized entries | Sized bytes | Verified extensions & contents |
|---|---|---|---|---|---|---|
| Total | 313 | 100.0% | 311 | 2 | 3,847,708 | Exact byte parity (3.85 MB; shares rounded to 0.1%). |

Figure 5 divides all **313** entries into six mutually exclusive clusters designated `CL-01` through `CL-06` (`232 + 33 + 27 + 10 + 9 + 2 = 313`). Web API recon (`CL-06`, **232** items) holds **74.1%** of entries and **40.6%** of bytes. Together with escalation helpers (**29.9%** of bytes) and `bore` tunnels (**28.9%** of bytes), three clusters hold **99.3%** of workspace bytes.

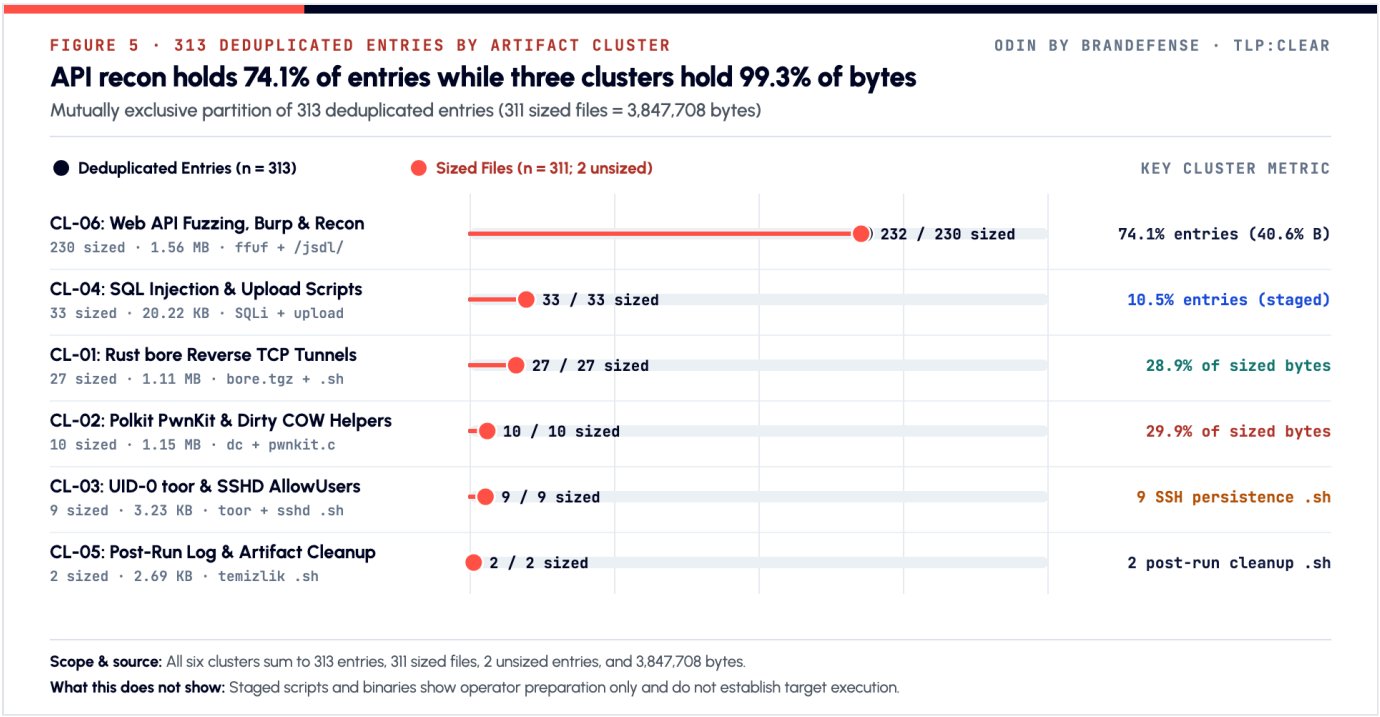


Figure 5: API recon holds 74.1% of entries while three clusters hold 99.3% of bytes

## Observation Dates Versus Iterative Script Revisions

Odin by Brandefense recorded `Workbench A` between `13 November 2026` and `26 November 2026` (`22:09 UTC`). Numbered revisions across the **258** shell scripts, including `db_data4.sh` and

`bore_restart3.sh`, reflect iterative staging on `Workbench A`.

Those revisions do not measure runtime duration or external impact. Without target-side logs, none of the **313** workspace entries confirm execution on the scoped regional messaging gateway.

## 3. Web API Fuzzing, Captured HTTP Templates, and SQL/Upload Probes

Messaging and dealer gateways expose HTTP endpoints for bulk dispatch, account credits, delivery reports, and file imports. Figure 3 shows an analytical model of five staged steps on `Workbench A`. Every step uses dashed arrows to show staged preparation rather than verified target execution.

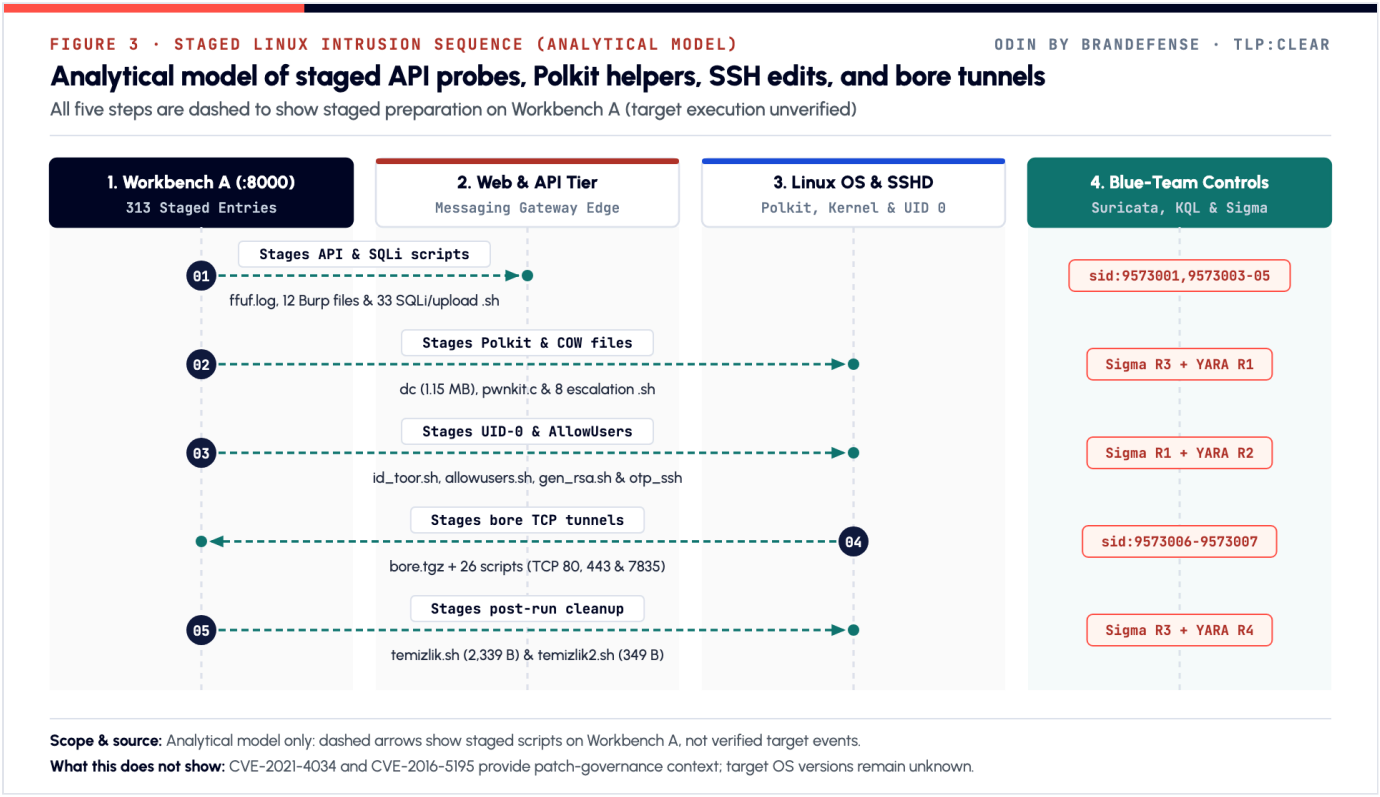


Figure 3: Analytical model of staged API probes, Polkit helpers, SSH edits, and bore tunnels

## Endpoint Discovery, API Schema Files, and HTTP Templates (`CL-06`, 232 Entries)

Cluster `CL-06` holds **232** entries (**230** sized files totaling **1,560,301** bytes plus **2** unsized root items) scoping a regional messaging gateway whose name and domain are withheld. Root holds

`api_doc.pdf` (**635,631** B), extracted text `api_doc.txt` (**22,693** B), and parameter list `plist.txt` (**20,030** B). Those files document endpoint paths, authentication headers, and request schemas.

Six scanner logs and URL lists total **366,636** bytes in `(root)`. Stored log `ffuf.log` (**293,402** B) sits beside `arjun.log` (**25,865** B) and `gau_urls.txt` (**16,840** B). Three more files cover `httpx_install.log` (**14,937** B), `gau_paths_normalized.txt` (**14,275** B), and `katana.log` (**1,317** B).

Two subdirectories store repeatable request and frontend files. Folder `/burp_requests/` holds **12** HTTP templates (**2,461** B), built by `make_burp_requests.sh` (**918** B; **3,379** B combined across **13** items). Folder `/jsdl/` holds **7** JavaScript files (**318,298** B), led by `app.js` (**111,747** B), `jquery.js` (**96,763** B), and `jquery.min.js` (**94,840** B).

## Staged SQL Injection and Multipart Upload Validation Scripts ( `CL-04` , 33 Entries)

Cluster `CL-04` (**33** sized scripts totaling **20,215** bytes, **10.5%** of entries) tests database inputs and file-upload routes. Numeric IDs and pagination parameters expose SQL tables when backend code concatenates unquoted integers instead of binding typed parameters. Storing uploads on object buckets or `noexec` mounts outside PHP interpreter paths blocks script execution even if extension filters fail.

| `CL-04` script subgroup | Representative staged artifacts | Defensive control focus |
|---|---|---|
| **Numeric SQL & schema probes** | `db_data4.sh` (1,016 B), `sms_test2.sh` (995 B), `qsql.sh` (855 B), `numsqli.sh` (829 B), `sqli_scan.sh` , `mysql_try.sh` , `dbadmin.sh` / `2.sh` , `db_data.sh` / `2.sh` , `smsschema.sh` . | Bind typed SQL parameters on all numeric fields and alert via `sid:9573003` –9573004. |
| **Multipart upload checks** | `admin_upload.sh` (1,427 B), `upload_final.sh` (908 B), `upload_try.sh` , `phptest.sh` , `loop_payload.sh` , 1 root `.php` (356 B), plus 2 withheld target-suffixed scripts (1,059 B and 986 B; 2,045 B). | Enforce strict upload extension allowlists, mount upload paths `noexec` , and alert via `sid:9573005` . |

## 4. Staged Polkit/Dirty COW Escalation, UID-0 SSH Persistence, and Bore Tunnels

Alongside web API and SQL probes, `Workbench A` holds **48** Linux post-exploitation, OpenSSH, tunnel, and cleanup files (**2,267,192** bytes, **58.9%** of sized bytes). Figure 4 shows the `TCP 8000` directory layout across `(root)`, `/burp_requests/`, and `/jsdl/`.




Figure 4: One open TCP 8000 directory spans 294 root items and two recon folders

## CISA KEV Patch Baselines: Polkit `CVE-2021-4034` and Dirty COW `CVE-2016-5195` (`CL-02`, 10 Entries)

Cluster `CL-02` (**10** sized items totaling **1,150,699** bytes, **29.9%** of sized bytes) stages local Linux privilege-escalation helpers (`T1068`; `[REF-06]`). Root holds compiled binary `/dc` (**1,146,296** B), C source `/pwnkit.c` (**1,267** B), and **8** shell scripts (**3,136** B).

Those **8** wrappers start with `pk_root_run.sh` (**932** B), `root_run.sh` (**926** B), `pk_go.sh` (**350** B), and `suid.sh` (**263** B). Four smaller scripts cover `cow_restart.sh` (**247** B), `pk_test.sh` (**226** B), `pwnkit.sh`, and `pk_dbg.sh`.

Both flaws sit in CISA's catalog of known exploited vulnerabilities ( [REF-03] ). Under NVD record CVE-2021-4034 ( [REF-01] ), Polkit's pkexec utility was vulnerable in polkit versions up to 0.120 before January 2022 vendor updates. Out-of-bounds environment handling when invoked with zero arguments let local users gain root privileges.

Under NVD record CVE-2016-5195 ( [REF-02] ), Dirty COW affected mm/gup.c across Linux kernel 2.x through 4.x before 4.8.3. Improper copy-on-write handling of read-only memory mappings let local users escalate privileges. Operators still bundle both helpers for unmaintained Linux appliances; both CVEs serve strictly as patch-baseline context.

## Auditing UID-0 toor Accounts and OpenSSH AllowUsers Directives ( CL-03 , 9 Entries)

Cluster CL-03 (9 sized .sh scripts totaling 3,229 bytes) stages UID-0 accounts and OpenSSH edits ( T1136.001 , T1098.004 ; [REF-05] , [REF-07] , [REF-08] ):

| Staged script in CL-03 | Size (bytes) | Defensive audit surface |
|---|---|---|
| persist.sh | 932 | Background persistence and session maintenance wrapper. |
| otp_ssh.sh | 795 | SSH authentication and PAM or one-time token policy check script. |
| allowusers.sh | 379 | Edits /etc/ssh/sshd_config AllowUsers directives ( [REF-05] ) to permit added accounts. |
| sshd_diag.sh | 346 | Inspects OpenSSH daemon settings, active listeners, and login rules ( [REF-05] ). |
| setsid_test.sh | 311 | Detaches background sessions ( setsid ) from web worker process groups. |
| rsh.sh | 165 | Remote shell helper staged beside SSH configuration scripts. |
| gen_rsa.sh | 123 | Generates SSH RSA key pairs for authorized_keys persistence ( [REF-08] ). |

| Staged script in `CL-03` | Size (bytes) | Defensive audit surface |
|---|---|---|
| `id_toor2.sh` | 117 | Second revision of secondary UID-0 `toor` account setup ( `[REF-07]` ). |
| `id_toor.sh` | 61 | Initial script staging a secondary UID-0 `toor` local account ( `[REF-07]` ). |
| Cluster `CL-03` total | 3,229 | Exact byte total across all 9 SSH persistence scripts. |

Per `sshd_config(5)` ( `[REF-05]` ), an explicit `AllowUsers` directive blocks every login account omitted from that allowlist, including UID-0 accounts. OpenSSH checks UID 0 when enforcing `PermitRootLogin no`, so a secondary UID-0 username ( `toor` ) is also denied SSH login despite appearing in `AllowUsers`. Defenders must monitor `/etc/passwd` for non- `root` UID-0 rows and audit effective OpenSSH policy with `sshd -T`.

## Egress Control: Rust `bore` Tunnels and Cleanup Scripts ( `CL-01` & `CL-05` , 29 Entries)

Cluster `CL-01` (**27** sized entries totaling **1,110,576** bytes, **28.9%** of sized bytes) stages reverse TCP relays ( `T1572` ; `[REF-04]` , `[REF-09]` ). Archive `/bore.tgz` (**1,097,869** B) packages `ekzhang/bore` ( `[REF-04]` ), a single-binary Rust TCP tunnel client. Twenty-six shell scripts (**12,707** B) wrap tunnel deployment, port selection, and restarts across **14** core `bore` wrappers and **12** relay helpers.

Standard `ekzhang/bore` frames control handshakes as null-delimited JSON ( `{"Hello":<port>}` ) over cleartext TCP ( `[REF-04]` ). Default `bore` uses `TCP 7835` , whereas `bore80.sh` and `bore443.sh` target `TCP 80` and **443** to bypass port-only egress filters.

Cluster `CL-05` (**2** sized scripts totaling **2,688** B) stages `temizlik.sh` (**2,339** B) and `temizlik2.sh` (**349** B) for post-run cleanup ( `T1070.004` ; `[REF-10]` ). Figure 2 maps all six clusters to defensive controls.

| Tunnel or cleanup group | Count | Sized bytes | Verified `Workbench A` artifacts |
|---|---|---|---|
| Rust `bore` archive ( `CL-01` ) | 1 | 1,097,869 | `/bore.tgz` (1,097,869 B) packaging the `ekzhang/bore` TCP tunnel binary ( `[REF-04]` ). |
| Tunnel & relay `.sh` wrappers ( `CL-01` ) | 26 | 12,707 | 14 core `bore` , `tunnel2.sh` (810 B), and `portproxy.sh` wrappers ( `bore443.sh` 1,043 B, `bore_vhost2.sh` 979 B, `bore80.sh` ) plus 12 relay scripts ( `relay_shell.sh` , `root_tunnel.sh` , `bore_vhost.sh` ). |
| Post-run cleanup scripts ( `CL-05` ) | 2 | 2,688 | `temizlik.sh` (2,339 B) and `temizlik2.sh` (349 B) targeting temporary files and host logs ( `[REF-10]` ). |

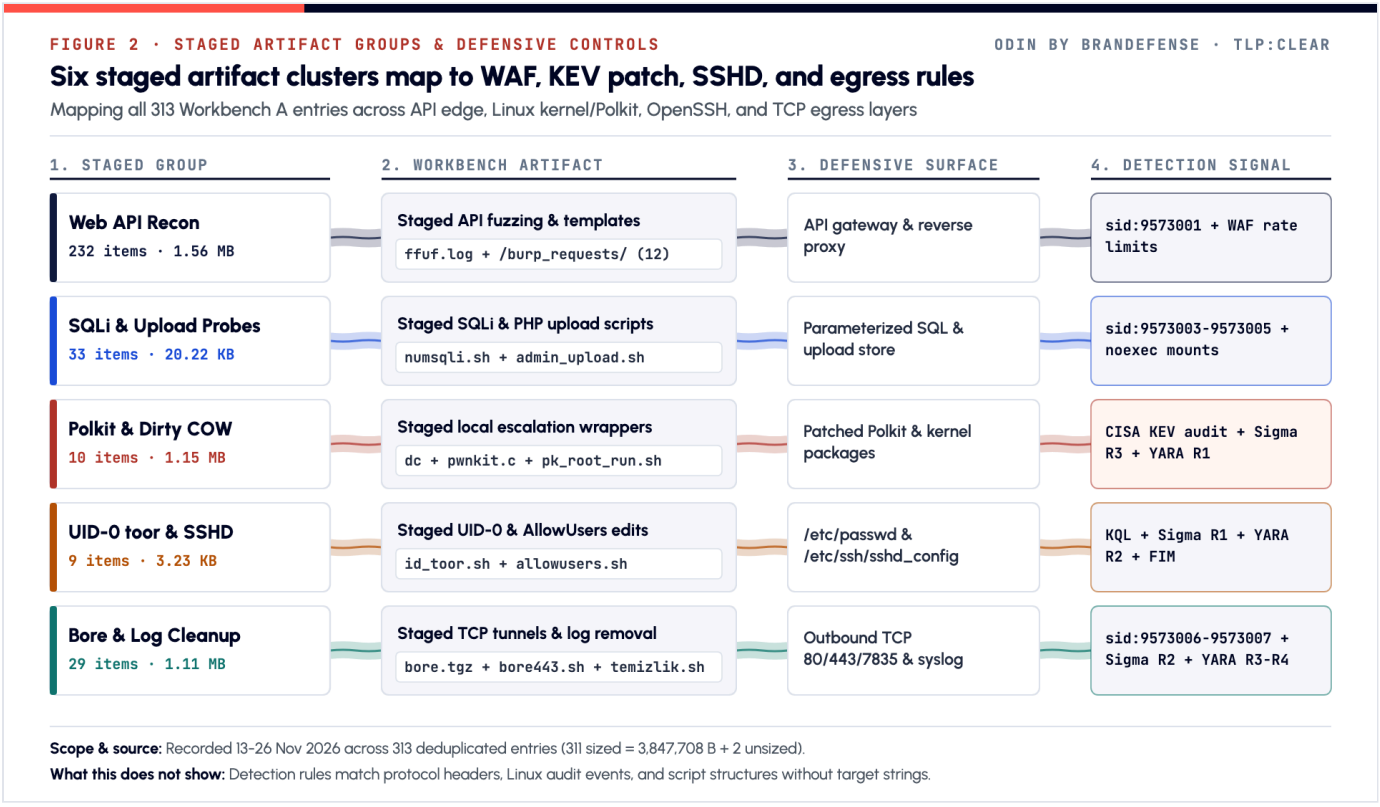


Figure 2: Six staged artifact clusters map to WAF, KEV patch, SSHD, and egress rules

## 5. Enterprise API, Linux Patch, OpenSSH, and Egress Exposure Analysis

Blue teams should verify five defensive controls mapped to `Workbench A` before assuming external impact occurred:

| Exposure layer | Observed `Workbench A` evidence | Control to verify | Operational risk |
| --- | --- | --- | --- |
| **Web API docs & fuzzing** | 232 `CL-06` entries (1.56 MB), `api_doc.pdf` (635,631 B), `ffuf.log` (293,402 B), 12 `/burp_requests/` files. | Verify whether API specs require login and gateways enforce rate limits and schema checks. | Exposed specs and unthrottled fuzzing reveal hidden parameters and legacy dealer routes. |
| **SQL & upload handlers** | 33 `CL-04` scripts (20,215 B): `numsqli.sh`, `qsql.sh`, `smsschema.sh`, `admin_upload.sh`, `upload_final.sh`; execution unverified. | Verify whether SQL queries bind typed parameters and uploads sit on `noexec` storage. | Numeric SQL flaws expose tables, while executable upload paths allow web-worker code execution. |
| **Polkit & kernel KEV baseline** | 10 `CL-02` items (1,150,699 B): `/dc` (1.15 MB), `pwnkit.c` (1,267 B), `pk_root_run.sh`, `cow_restart.sh`; execution unverified. | Verify whether `polkit` carries January 2022 vendor patches (`<= 0.120`, `CVE-2021-4034`), each kernel matches its vendor's `CVE-2016-5195` fixed package, and unsupported releases are upgraded (`[REF-01]`, `[REF-02]`, `[REF-03]`). | Unpatched KEV flaws let low-privilege web accounts escalate to UID 0 (`T1068`; `[REF-06]`). |

| Exposure layer | Observed `Workbench A` evidence | Control to verify | Operational risk |
|---|---|---|---|
| **UID-0 & `sshd_config`** | 9 `CL-03` scripts (3,229 B): `id_toor.sh`, `id_toor2.sh`, `allowusers.sh`, `sshd_diag.sh`, `gen_rsa.sh`, `otp_ssh.sh`. | Verify whether FIM alerts on non-`root` UID-0 accounts, `authorized_keys`, and `AllowUsers` edits (`[REF-05]`). | A secondary UID-0 `toor` account is root-equivalent locally, but `PermitRootLogin no` still blocks its SSH login (`T1136.001`, `T1098.004`). |
| **TCP egress & host logs** | 27 `CL-01` `bore` items (1,110,576 B; `bore80.sh`, `bore443.sh`) and 2 `CL-05` cleanup scripts (2,688 B). | Verify whether servers block direct TCP egress on 80, 443, and 7835 and ship `auth.log` / `auditd` off-host (`[REF-04]`, `[REF-10]`). | Raw TCP `bore` relays expose SSH (`T1572`), while local log wiping erases host trails (`T1070.004`). |

## 6. Alternative Hypotheses

| Hypothesis | Evidence for | Evidence against | Assessment |
|---|---|---|---|
| **Staged scripts achieved root access on the gateway** | Scripts iterate across `id_toor.sh` / `id_toor2.sh`, `bore_restart.sh` – `3.sh`, `db_data.sh` – `4.sh`, and `temizlik.sh`. | `Workbench A` holds no target `/etc/shadow` files, target `sshd_config` outputs, or target logs confirming execution. | **Unverified.** Treat all 313 entries strictly as staged preparation on `Workbench A`. |
| **Legacy `CVE-2016-5195` and `CVE-2021-4034` files reflect an old archive** | `CVE-2016-5195` dates to 2016 and `CVE-2021-4034` dates to January 2022 (`[REF-01]`, `[REF-02]`). | The files shared a directory available in November; their creation and execution dates are unknown. | **Unresolved.** The live listing establishes November availability, but the package lacks file creation or execution dates that would rule out an older archive. |

| Hypothesis | Evidence for | Evidence against | Assessment |
| --- | --- | --- | --- |
| **Benign developer host or authorized test server** | Holds standard recon tools (`ffuf`, `arjun`, `httpx`, `katana`) and API PDF docs. | Co-stages UID-0 `toor` accounts, `AllowUsers` edits, TCP 80/443 `bore` relays, `temizlik.sh` log wiping, and open `TCP 8000` templates. | **Contradicted.** Covert UID-0 SSH scripts, log wiping, and open staging confirm an adversary workbench. |
| **Adversary workbench pairing gateway recon with reusable Linux scripts** | All 6 clusters (`API recon`, `SQLi/upload`, `privesc`, `UID-0 SSH`, `bore`, `cleanup`) sit together on `Workbench A`. | Generic Linux scripts (`CL-01`–`CL-03`, `CL-05`) can also be reused across other Linux targets. | **Supported.** Assess `Workbench A` as an adversary workbench combining gateway recon with modular Linux scripts. |

## 7. Validated Detection Rules

Zero IPv4 addresses and zero file SHA-256 digests are published. Blue teams should deploy **protocol, Linux process, OpenSSH policy, and file-structure detections**. Static checks passed in the Odin Suricata linter (**0** problems across **6** rules) and structural checks across KQL, **3** Sigma rules, and **4** `filesize`-bounded YARA rules. Because live `suricata -T`, `Kusto.Language`, `pySigma`, and `YARA` binaries were not run locally, teams should test each rule in staging first.

### 7.1 Suricata 7.x Network Signatures (`sid:9573001`, `sid:9573003`–9573007)

Rule `sid:9573001` detects inbound `Fuzz Faster U Fool` (`ffuf`) HTTP scanner headers. Rules `sid:9573003`–**9573005** alert on inbound `INFORMATION_SCHEMA.` SQL probes in URIs (`SELECT`) and bodies (`UNION`) plus multipart `.php` upload posts.

Outbound rule `sid:9573006` alerts on HTTP `GET` requests ending in `/bore.tgz`. Rule `sid:9573007` matches outbound cleartext Rust `bore` JSON-null control handshakes (`{"Hello": <port>}`; `[REF-04]`) over `TCP 80`, **443**, or **7835**. None of these six `alert` rules block traffic inline. The five HTTP rules (`sid:9573001`, `sid:9573003`–**9573006**) need cleartext HTTP or proxy decryption, while `sid:9573007` inspects raw `bore` TCP streams on **80**, **443**, and **7835**.

```
SURICATA

# Odin by Brandefense - Monograph 73 Suricata 7.x Ruleset
# Static syntax checked with Odin Suricata linter (0 problems); test with suricata -T
before deployment.

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound ffuf Web API Fuzz
ing User-Agent"; flow:established,to_server; http.user_agent; content:"Fuzz Faster U F
ool"; fast_pattern; classtype:web-application-attack; sid:9573001; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound SQLi INFORMATION_
SCHEMA URI Probe"; flow:established,to_server; http.uri; content:"INFORMATION_SCHEM
A."; nocase; fast_pattern; content:"SELECT"; nocase; classtype:web-application-attack;
sid:9573003; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound SQLi INFORMATION_
SCHEMA Body Probe"; flow:established,to_server; http.request_body; content:"INFORMATIO
N_SCHEMA."; nocase; fast_pattern; content:"UNION"; nocase; classtype:web-application-a
ttack; sid:9573004; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound Multipart PHP Upl
oad Probe"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.req
uest_body; content:"Content-Disposition: form-data"; nocase; fast_pattern; content:".p
hp"; nocase; classtype:web-application-attack; sid:9573005; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R73 Outbound HTTP Fetch of bo
re.tgz Archive"; flow:established,to_server; http.method; content:"GET"; bsize:3; htt
p.uri; content:"/bore.tgz"; endswith; fast_pattern; classtype:trojan-activity; sid:957
3006; rev:1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET [80,443,7835] (msg:"ODIN-R73 Outbound Rust bo
re TCP Control Hello Frame"; flow:established,to_server; content:"{|22|Hello|22|:"; fa
st_pattern; content:"}|00|"; distance:0; within:8; classtype:policy-violation; sid:957
3007; rev:1;)
```

## 7.2 Microsoft Sentinel KQL Hunt

Replace the two placeholder FQDNs in `ApprovedAdminAndBastionHosts` with your approved Linux bastion and security audit hosts before scheduling.

```
KQL

// Odin by Brandefense - Monograph 73
// Hunts Linux UID-0 toor account creation, sshd_config AllowUsers edits, Rust bore tu
nnel processes, and unexpected daemon egress.
let Lookback = 7d;
let ApprovedAdminAndBastionHosts = dynamic(["linux-bastion-01.corp.example", "sec-audi
t-01.corp.example"]);
let LinuxPrivescSshAndBoreProcessEvents =
    DeviceProcessEvents
```

```
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminAndBastionHosts))
    | where (
        (FileName in~ ("useradd", "usermod", "chpasswd") and ProcessCommandLine has "t
oor")
        or (FileName in~ ("sed", "tee", "awk", "cp", "mv") and ProcessCommandLine has
"/etc/ssh/sshd_config" and ProcessCommandLine has_any ("AllowUsers", "PermitRootLogi
n"))
        or (FileName =~ "bore" and ProcessCommandLine has "local" and ProcessCommandLi
ne has "--to")
        or (ProcessCommandLine has_any ("bore.tgz", "pwnkit", "cow_restart", "id_too
r", "allowusers.sh", "temizlik"))
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "LinuxSshPersistenceOrB
oreProcess", Detail = strcat(FileName, ": ", ProcessCommandLine);
let UnexpectedLinuxTunnelPublicEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminAndBastionHosts))
    | where RemoteIPType == "Public"
    | where (
        InitiatingProcessFileName =~ "bore"
        or (InitiatingProcessFileName in~ ("sshd", "php-fpm", "apache2", "nginx") and
RemotePort in (80, 443, 7835, 8000))
    )
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, Signa
lType = "UnexpectedLinuxTunnelOrDaemonEgress", Detail = strcat(InitiatingProcessFileNa
me, " -> ", RemoteIP, ":", tostring(RemotePort));
union LinuxPrivescSshAndBoreProcessEvents, UnexpectedLinuxTunnelPublicEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count
(), Signals = make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceNa
me, AccountName
| order by LastSeen desc
```

## 7.3 Portable Sigma and YARA Rules

The three Sigma rules hunt Linux process creation outside the named administration hosts. Rule 1 matches account or SSH policy commands that reference `toor`, `AllowUsers`, or the staged script names. Rule 2 matches `bore` launches, archive fetches, and wrapper names. Rule 3 matches Polkit or Dirty COW helpers and cleanup scripts.

```yaml
title: Linux Secondary UID-0 Account or SSHD AllowUsers Modification
id: 73a4d912-5c38-4e21-8b44-73c10f92a301
status: experimental
description: Detects Linux account utilities or text processors referencing a secondar
y toor account or modifying OpenSSH AllowUsers and persistence scripts outside configu
ration management hosts.
author: Odin by Brandefense
date: 2026-11-27
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith:
      - '/useradd'
      - '/usermod'
      - '/chpasswd'
      - '/sed'
      - '/tee'
      - '/sh'
      - '/bash'
  selection_args:
    CommandLine|contains:
      - 'toor'
      - 'AllowUsers'
      - 'id_toor'
      - 'allowusers.sh'
      - 'otp_ssh.sh'
  filter_cfg_hosts:
    ComputerName:
      - 'cfg-mgmt-01.corp.example'
  condition: selection_bin and selection_args and not filter_cfg_hosts
falsepositives:
  - Approved configuration management runs updating OpenSSH AllowUsers baselines on al
lowlisted administration hosts
level: high
tags:
  - attack.persistence
  - attack.t1136.001
  - attack.t1098.004
---
title: Linux Rust Bore Reverse TCP Tunnel Execution or Archive Fetch
id: 73a4d912-5c38-4e21-8b44-73c10f92a302
status: experimental
description: Detects execution of the Rust bore TCP tunnel client, bore wrapper script
```

```
s, or bore.tgz archive retrieval outside designated developer tunnel gateways.
author: Odin by Brandefense
date: 2026-11-27
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith:
      - '/bore'
      - '/sh'
      - '/bash'
      - '/tar'
      - '/curl'
      - '/wget'
  selection_cli:
    CommandLine|contains:
      - 'bore local '
      - 'bore.tgz'
      - 'bore80.sh'
      - 'bore443.sh'
      - 'bore_vhost'
      - 'bore_restart'
  filter_tunnel_hosts:
    ComputerName:
      - 'tunnel-gw-01.corp.example'
  condition: selection_bin and selection_cli and not filter_tunnel_hosts
falsepositives:
  - Authorized developer local port forwarding on approved sandbox gateways
level: high
tags:
  - attack.command-and-control
  - attack.t1572
---
title: Linux Polkit or DirtyCow Escalation Wrapper and Post-Run Cleanup
id: 73a4d912-5c38-4e21-8b44-73c10f92a303
status: experimental
description: Detects Linux command lines referencing staged Polkit CVE-2021-4034 wrapp
ers, Dirty COW CVE-2016-5195 helpers, or post-run cleanup scripts outside isolated aud
it hosts.
author: Odin by Brandefense
date: 2026-11-27
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
```

```
    Image|endswith:
      - '/sh'
      - '/bash'
      - '/pkexec'
      - '/gcc'
      - '/cc'
  selection_markers:
    CommandLine|contains:
      - 'pwnkit'
      - 'pk_root_run'
      - 'pk_go.sh'
      - 'cow_restart'
      - 'temizlik'
  filter_audit_hosts:
    ComputerName:
      - 'sec-audit-01.corp.example'
  condition: selection_bin and selection_markers and not filter_audit_hosts
falsepositives:
  - Authorized kernel and Polkit patch validation inside isolated security test hosts
level: high
tags:
  - attack.privilege-escalation
  - attack.t1068
  - attack.defense-evasion
  - attack.t1070.004
```

Four `filesize`-bounded YARA rules in `odin_pwnkit_uid0_ssh_and_bore_artifacts.yar` detect Polkit and Dirty COW wrappers (**< 32KB**) alongside UID-0 `toor` and `AllowUsers` scripts (**< 16KB**). Two companion rules (**< 16KB**) match Rust `bore` tunnel wrappers and Linux log-cleanup scripts.

```
YARA
/*
   Odin by Brandefense - Monograph 73 YARA Suite
   4 filesize-bounded structural rules covering Polkit/Dirty COW escalation wrappers,
   UID-0 toor and OpenSSH AllowUsers persistence scripts, Rust bore TCP tunnel wrapper
s,
   and post-run Linux log cleanup scripts (zero victim or vendor identifiers).
*/

rule ODIN_R73_Polkit_or_DirtyCow_Wrapper_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
```

```
        description = "Detects staged Linux shell wrappers referencing Polkit pkexec C
VE-2021-4034 or Dirty COW CVE-2016-5195 markers"
        date = "2026-11-27"
    strings:
        $sh1 = "#!/bin/sh" ascii
        $sh2 = "#!/bin/bash" ascii
        $p1 = "pkexec" ascii
        $p2 = "GCONV_PATH" ascii
        $p3 = "pwnkit" ascii nocase
        $c1 = "/proc/self/mem" ascii
        $c2 = "cow_restart" ascii nocase
    condition:
        filesize < 32KB and ($sh1 or $sh2) and (2 of ($p*) or ($c1 and $c2))
}

rule ODIN_R73_UID0_Toor_and_SSHD_AllowUsers_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects scripts staging a secondary toor account or changing Op
enSSH sshd_config AllowUsers"
        date = "2026-11-27"
    strings:
        $u1 = "toor" ascii
        $u2 = "/etc/passwd" ascii
        $u3 = "useradd" ascii
        $s1 = "sshd_config" ascii
        $s2 = "AllowUsers" ascii
        $s3 = "authorized_keys" ascii
    condition:
        filesize < 16KB and (($u1 and ($u2 or $u3)) or ($s1 and $s2))
}

rule ODIN_R73_Rust_Bore_TCP_Tunnel_Wrapper
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects shell wrappers launching the Rust bore TCP tunnel binar
y or unpacking bore.tgz"
        date = "2026-11-27"
    strings:
        $b1 = "bore local" ascii
        $b2 = "bore.tgz" ascii
        $t1 = "--to " ascii
        $t2 = "--port" ascii
```

```
        $t3 = "setsid" ascii
        $t4 = "nohup" ascii
    condition:
        filesize < 16KB and ($b1 or $b2) and $t1 and 1 of ($t2, $t3, $t4)
}

rule ODIN_R73_PostRun_Log_and_History_Cleanup_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects post-run Linux shell scripts targeting authentication l
ogs, shell history variables, and temporary staging paths"
        date = "2026-11-27"
    strings:
        $sh1 = "#!/bin/" ascii
        $l1 = "/var/log/auth.log" ascii
        $l2 = "/var/log/secure" ascii
        $l3 = ".bash_history" ascii
        $l4 = "/var/log/wtmp" ascii
        $c1 = "HISTFILE" ascii
        $c2 = "/tmp/" ascii
    condition:
        filesize < 16KB and $sh1 and 2 of ($l*) and $c1 and $c2
}
```

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

Zero IPv4 addresses or file SHA-256 digests appear in the indicator feeds. `Workbench A` holds target-specific HTTP request templates and upload script names that are withheld under victim-protection rules. Defenders can hunt using the nine behavioral and host markers below.

| Marker category | Pattern / protocol or host artifact | Layer | Defensive hunt & control |
|---|---|---|---|
| API fuzzing header | `User-Agent: Fuzz Faster U Fool` (`ffuf.log`, 293,402 B) | `TCP 443` / HTTP | Rate-limit API fuzzing at the WAF edge and alert via `sid:9573001`. |
| SQL catalog probe | `INFORMATION_SCHEMA.` + `SELECT` / `UNION` (`numsqli.sh`, `smsschema.sh`) | `TCP 443` / HTTP | Enforce parameterized SQL queries and alert via `sid:9573003`–9573004. |
| PHP upload check | `POST` `Content-Disposition: form-data` + `.php` (`admin_upload.sh`) | `TCP 443` / HTTP | Enforce upload allowlists, mount upload paths `noexec`, and alert via `sid:9573005`. |
| Polkit & COW wrappers | `pkexec` / `GCONV_PATH` / `pwnkit` & `cow_restart` (10 `CL-02` items) | Linux local OS | Verify January 2022 `polkit` fixes and vendor `CVE-2016-5195` fixed kernel packages (`[REF-01]`, `[REF-02]`), Sigma Rule 3, YARA Rule 1. |
| Duplicate UID-0 account | Secondary UID-0 `toor` entry in `/etc/passwd` (`id_toor.sh`, `id_toor2.sh`) | `TCP 22` / Linux | Use FIM to confirm non-`root` UID-0 accounts (`[REF-07]`); KQL and Sigma Rule 1 hunt related commands, while YARA Rule 2 hunts staged scripts. |

| Marker category | Pattern / protocol or host artifact | Layer | Defensive hunt & control |
|---|---|---|---|
| **OpenSSH policy edit** | `/etc/ssh/sshd_config` `AllowUsers` & `authorized_keys` edits ( `allowusers.sh` ) | `TCP 22` / SSHD | Audit `sshd -T` policy baselines ( `[REF-05]` , `[REF-08]` ) and alert via FIM, KQL, and Sigma Rule 1. |
| **Tunnel archive fetch** | Outbound HTTP `GET /bore.tgz` (1,097,869 B archive + `get_bore.sh` ) | `TCP 8000` egress | Enforce outbound HTTP proxy allowlists and alert via `sid:9573006` . |
| **`bore` TCP handshake** | `{"Hello":<port>}x00` cleartext frame ( `bore80.sh` , `bore443.sh` ) | `TCP 80/443/7835` | Block direct server TCP egress ( `[REF-04]` , `[REF-09]` ) and alert via `sid:9573007` , KQL, Sigma Rule 2. |
| **Post-run log cleanup** | `HISTFILE` suppression & `auth.log` / `wtmp` removal ( `temizlik.sh` , 2,339 B) | Linux syslog | Stream `auth.log` and `auditd` to append-only SIEM ( `[REF-10]` ), Sigma Rule 3, and YARA Rule 4. |

## 9. Hardening Priorities for Linux Hosts, OpenSSH Daemons, and Egress Gateways

Linux platform, application, and network defenders should execute five controls in priority order:

1. **Verify CISA KEV Polkit and Linux kernel baselines ( `0-48 Hours` ).** Confirm DMZ and web Linux servers carry January 2022 `polkit` fixes ( `<= 0.120` , `CVE-2021-4034` ; `[REF-01]` , `[REF-03]` ). Verify each kernel against its vendor's `CVE-2016-5195` fixed package, upgrade unsupported releases ( `[REF-02]` ), and remove unneeded SUID bits from `/usr/bin/pkexec` ( `[REF-06]` ).
2. **Enforce UID-0 `/etc/passwd` invariants and lock `sshd_config` ( `0-72 Hours` ).** Verify daily that `root` is the sole UID-0 user in `/etc/passwd` ( `[REF-07]` ). Audit OpenSSH settings via `sshd -T`

( `[REF-05]` ), set `PermitRootLogin no` , lock `AllowUsers` and `authorized_keys` ( `[REF-08]` ), and monitor `/etc/ssh/sshd_config` .

3. **Block unproxied outbound TCP tunnels from server subnets ( `Days 1-7` ).** Deny direct outbound TCP sessions from application servers to public hosts on `TCP 80` , **443**, **7835**, and **8000** ( `[REF-04]` , `[REF-09]` ). Requiring authenticated HTTP/TLS forward proxies stops raw TCP `bore` control channels on ports **80** and **443**.
4. **Harden messaging API schemas, SQL queries, and file uploads ( `Days 7-14` ).** Gate dealer and messaging API specs ( `api_doc.pdf` ) behind authentication and bind typed SQL parameters on numeric fields. Isolate uploads on `noexec,nosuid,nodev` mounts or external object storage.
5. **Stream authentication and `auditd` logs to append-only collectors ( `Days 14-30` ).** Ship `/var/log/auth.log` , `/var/log/secure` , and `auditd` events off-host in real time. Lock `auditd` rules ( `-e 2` ) so cleanup scripts ( `temizlik.sh` , `temizlik2.sh` ; `[REF-10]` ) cannot erase host records.

## 10. Scope, Limits, and External References

Odin by Brandefense analyzed `Workbench A` on `TCP 8000` of a Netherlands VPS. The open directory was first recorded on `13 November 2026` and last verified on `26 November 2026` at `22:09 UTC` . Deduplicated indexing captured all **313** reported entries across **311** sized files totaling **3,847,708** bytes (**3.85 MB**) and **2** unsized root items. Those entries span **294** root items, **12** `/burp_requests/` templates, and **7** `/jsdl/` assets, with **4** snapshot files stored in full.

All **313** entries represent staged preparation on `Workbench A` and do not confirm SQL extraction, local escalation, UID-0 account creation, or `bore` tunnel use on any target. Both `CVE-2021-4034` and `CVE-2016-5195` serve strictly as CISA KEV patch-governance context. `Workbench A` 's IPv4 address, all SHA-256 digests, **2** target-suffixed upload script names, and all messaging gateway identifiers are withheld.


1. `[REF-01]` NIST National Vulnerability Database, *CVE-2021-4034 Detail: Polkit pkexec Local Privilege Escalation* (verified on 27 November 2026).
2. `[REF-02]` NIST National Vulnerability Database, *CVE-2016-5195 Detail: Linux Kernel Copy-on-Write Dirty COW Privilege Escalation* (verified on 27 November 2026).
3. `[REF-03]` Cybersecurity and Infrastructure Security Agency (CISA), *Known Exploited Vulnerabilities Catalog* (verified on 27 November 2026).

4. [REF-04] Official Source Repository, *ekzhang/bore: Modern Simple TCP Tunnel in Rust* (verified on 27 November 2026).
5. [REF-05] OpenBSD Manual Pages, *sshd_config(5): OpenSSH Daemon AllowUsers and PermitRootLogin Directives* (verified on 27 November 2026).
6. [REF-06] MITRE ATT&CK, *Exploitation for Privilege Escalation (T1068)* (verified on 27 November 2026).
7. [REF-07] MITRE ATT&CK, *Create Account: Local Account (T1136.001)* (verified on 27 November 2026).
8. [REF-08] MITRE ATT&CK, *Account Manipulation: SSH Authorized Keys (T1098.004)* (verified on 27 November 2026).
9. [REF-09] MITRE ATT&CK, *Protocol Tunneling (T1572)* (verified on 27 November 2026).
10. [REF-10] MITRE ATT&CK, *Indicator Removal: File Deletion (T1070.004)* (verified on 27 November 2026).

---

## BRANDEFENSE CAPABILITIES RELATED TO THIS RESEARCH

### Cyber Threat Intelligence

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

### Vulnerability Intel

Prioritise the flaws that attackers actually stage.

### Threat Search

Pivot on an indicator across the Brandefense dataset.

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.

9 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.

YARA **odin_pwnkit_uid0_ssh_and_bore_artifacts.yar** 81 lines

```
/*
   Odin by Brandefense - Monograph 73 YARA Suite
   4 filesize-bounded structural rules covering Polkit/Dirty COW escalation wrappers,
   UID-0 toor and OpenSSH AllowUsers persistence scripts, Rust bore TCP tunnel wrapper
s,
   and post-run Linux log cleanup scripts (zero victim or vendor identifiers).
*/

rule ODIN_R73_Polkit_or_DirtyCow_Wrapper_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects staged Linux shell wrappers referencing Polkit pkexec C
VE-2021-4034 or Dirty COW CVE-2016-5195 markers"
        date = "2026-11-27"
    strings:
        $sh1 = "#!/bin/sh" ascii
        $sh2 = "#!/bin/bash" ascii
        $p1 = "pkexec" ascii
        $p2 = "GCONV_PATH" ascii
        $p3 = "pwnkit" ascii nocase
        $c1 = "/proc/self/mem" ascii
        $c2 = "cow_restart" ascii nocase
    condition:
        filesize < 32KB and ($sh1 or $sh2) and (2 of ($p*) or ($c1 and $c2))
}

rule ODIN_R73_UID0_Toor_and_SSHD_AllowUsers_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects scripts staging a secondary toor account or changing Op
enSSH sshd_config AllowUsers"
        date = "2026-11-27"
    strings:
        $u1 = "toor" ascii
        $u2 = "/etc/passwd" ascii
```

```
        $u3 = "useradd" ascii
        $s1 = "sshd_config" ascii
        $s2 = "AllowUsers" ascii
        $s3 = "authorized_keys" ascii
    condition:
        filesize < 16KB and (($u1 and ($u2 or $u3)) or ($s1 and $s2))
}

rule ODIN_R73_Rust_Bore_TCP_Tunnel_Wrapper
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects shell wrappers launching the Rust bore TCP tunnel binar
y or unpacking bore.tgz"
        date = "2026-11-27"
    strings:
        $b1 = "bore local" ascii
        $b2 = "bore.tgz" ascii
        $t1 = "--to " ascii
        $t2 = "--port" ascii
        $t3 = "setsid" ascii
        $t4 = "nohup" ascii
    condition:
        filesize < 16KB and ($b1 or $b2) and $t1 and 1 of ($t2, $t3, $t4)
}

rule ODIN_R73_PostRun_Log_and_History_Cleanup_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "73"
        description = "Detects post-run Linux shell scripts targeting authentication l
ogs, shell history variables, and temporary staging paths"
        date = "2026-11-27"
    strings:
        $sh1 = "#!/bin/" ascii
        $l1 = "/var/log/auth.log" ascii
        $l2 = "/var/log/secure" ascii
        $l3 = ".bash_history" ascii
        $l4 = "/var/log/wtmp" ascii
        $c1 = "HISTFILE" ascii
        $c2 = "/tmp/" ascii
    condition:
        filesize < 16KB and $sh1 and 2 of ($l*) and $c1 and $c2
}
```

Suricata **odin_pwnkit_uid0_ssh_and_bore_tunnels_hunt.rules** 10 lines

```
# Odin by Brandefense - Monograph 73 Suricata 7.x Ruleset
# Static syntax checked with Odin Suricata linter (0 problems); test with suricata -T
before deployment.

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound ffuf Web API Fuzz
ing User-Agent"; flow:established,to_server; http.user_agent; content:"Fuzz Faster U F
ool"; fast_pattern; classtype:web-application-attack; sid:9573001; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound SQLi INFORMATION_
SCHEMA URI Probe"; flow:established,to_server; http.uri; content:"INFORMATION_SCHEM
A."; nocase; fast_pattern; content:"SELECT"; nocase; classtype:web-application-attack;
sid:9573003; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound SQLi INFORMATION_
SCHEMA Body Probe"; flow:established,to_server; http.request_body; content:"INFORMATIO
N_SCHEMA."; nocase; fast_pattern; content:"UNION"; nocase; classtype:web-application-a
ttack; sid:9573004; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R73 Inbound Multipart PHP Upl
oad Probe"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.req
uest_body; content:"Content-Disposition: form-data"; nocase; fast_pattern; content:".p
hp"; nocase; classtype:web-application-attack; sid:9573005; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R73 Outbound HTTP Fetch of bo
re.tgz Archive"; flow:established,to_server; http.method; content:"GET"; bsize:3; htt
p.uri; content:"/bore.tgz"; endswith; fast_pattern; classtype:trojan-activity; sid:957
3006; rev:1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET [80,443,7835] (msg:"ODIN-R73 Outbound Rust bo
re TCP Control Hello Frame"; flow:established,to_server; content:"{|22|Hello|22|:"; fa
st_pattern; content:"}|00|"; distance:0; within:8; classtype:policy-violation; sid:957
3007; rev:1;)
```

Show full rule

Sentinel KQL **sentinel_pwnkit_uid0_ssh_and_bore_tunnels_hunt.kql** 29 lines

```
// Odin by Brandefense - Monograph 73
// Hunts Linux UID-0 toor account creation, sshd_config AllowUsers edits, Rust bore tu
nnel processes, and unexpected daemon egress.
let Lookback = 7d;
let ApprovedAdminAndBastionHosts = dynamic(["linux-bastion-01.corp.example", "sec-audi
t-01.corp.example"]);
let LinuxPrivescSshAndBoreProcessEvents =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminAndBastionHosts))
```

```
    | where (
        (FileName in~ ("useradd", "usermod", "chpasswd") and ProcessCommandLine has "t
oor")
        or (FileName in~ ("sed", "tee", "awk", "cp", "mv") and ProcessCommandLine has
"/etc/ssh/sshd_config" and ProcessCommandLine has_any ("AllowUsers", "PermitRootLogi
n"))
        or (FileName =~ "bore" and ProcessCommandLine has "local" and ProcessCommandLi
ne has "--to")
        or (ProcessCommandLine has_any ("bore.tgz", "pwnkit", "cow_restart", "id_too
r", "allowusers.sh", "temizlik"))
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "LinuxSshPersistenceOrB
oreProcess", Detail = strcat(FileName, ": ", ProcessCommandLine);
let UnexpectedLinuxTunnelPublicEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminAndBastionHosts))
    | where RemoteIPType == "Public"
    | where (
        InitiatingProcessFileName =~ "bore"
        or (InitiatingProcessFileName in~ ("sshd", "php-fpm", "apache2", "nginx") and
RemotePort in (80, 443, 7835, 8000))
    )
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, Signa
lType = "UnexpectedLinuxTunnelOrDaemonEgress", Detail = strcat(InitiatingProcessFileNa
me, " -> ", RemoteIP, ":", tostring(RemotePort));
union LinuxPrivescSshAndBoreProcessEvents, UnexpectedLinuxTunnelPublicEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count
(), Signals = make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceNa
me, AccountName
| order by LastSeen desc
```

**Sigma** `sigma_linux_uid0_ssh_bore_and_privesc.yml` 112 lines

```
title: Linux Secondary UID-0 Account or SSHD AllowUsers Modification
id: 73a4d912-5c38-4e21-8b44-73c10f92a301
status: experimental
description: Detects Linux account utilities or text processors referencing a secondar
y toor account or modifying OpenSSH AllowUsers and persistence scripts outside configu
ration management hosts.
author: Odin by Brandefense
date: 2026-11-27
logsource:
  category: process_creation
  product: linux
detection:
```

```
  selection_bin:
    Image|endswith:
      - '/useradd'
      - '/usermod'
      - '/chpasswd'
      - '/sed'
      - '/tee'
      - '/sh'
      - '/bash'
  selection_args:
    CommandLine|contains:
      - 'toor'
      - 'AllowUsers'
      - 'id_toor'
      - 'allowusers.sh'
      - 'otp_ssh.sh'
  filter_cfg_hosts:
    ComputerName:
      - 'cfg-mgmt-01.corp.example'
  condition: selection_bin and selection_args and not filter_cfg_hosts
falsepositives:
  - Approved configuration management runs updating OpenSSH AllowUsers baselines on al
lowlisted administration hosts
level: high
tags:
  - attack.persistence
  - attack.t1136.001
  - attack.t1098.004
---
title: Linux Rust Bore Reverse TCP Tunnel Execution or Archive Fetch
id: 73a4d912-5c38-4e21-8b44-73c10f92a302
status: experimental
description: Detects execution of the Rust bore TCP tunnel client, bore wrapper script
s, or bore.tgz archive retrieval outside designated developer tunnel gateways.
author: Odin by Brandefense
date: 2026-11-27
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith:
      - '/bore'
      - '/sh'
      - '/bash'
      - '/tar'
      - '/curl'
```

```
      - '/wget'
  selection_cli:
    CommandLine|contains:
      - 'bore local '
      - 'bore.tgz'
      - 'bore80.sh'
      - 'bore443.sh'
      - 'bore_vhost'
      - 'bore_restart'
  filter_tunnel_hosts:
    ComputerName:
      - 'tunnel-gw-01.corp.example'
  condition: selection_bin and selection_cli and not filter_tunnel_hosts
falsepositives:
  - Authorized developer local port forwarding on approved sandbox gateways
level: high
tags:
  - attack.command-and-control
  - attack.t1572
---
title: Linux Polkit or DirtyCow Escalation Wrapper and Post-Run Cleanup
id: 73a4d912-5c38-4e21-8b44-73c10f92a303
status: experimental
description: Detects Linux command lines referencing staged Polkit CVE-2021-4034 wrapp
ers, Dirty COW CVE-2016-5195 helpers, or post-run cleanup scripts outside isolated aud
it hosts.
author: Odin by Brandefense
date: 2026-11-27
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith:
      - '/sh'
      - '/bash'
      - '/pkexec'
      - '/gcc'
      - '/cc'
  selection_markers:
    CommandLine|contains:
      - 'pwnkit'
      - 'pk_root_run'
      - 'pk_go.sh'
      - 'cow_restart'
      - 'temizlik'
  filter_audit_hosts:
```

```
    ComputerName:
      - 'sec-audit-01.corp.example'
  condition: selection_bin and selection_markers and not filter_audit_hosts
falsepositives:
  - Authorized kernel and Polkit patch validation inside isolated security test hosts
level: high
tags:
  - attack.privilege-escalation
  - attack.t1068
  - attack.defense-evasion
  - attack.t1070.004
```

