ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 42 / 30 · CALLBACK INFRASTRUCTURE

Commercial OAST Listeners, Blind-XSS Panels, and Web Recon Suites

Separating Blind-XSS Callbacks from Dual-Use Scanners and Header Collisions

How 3,744 OAST, blind-XSS, browser-hook, and web recon endpoints narrow to behavioral callback hunting after filtering 2,481 dual-use tools, 154 PHP 8.5 header collisions, and two single-signal ezXSS rows

Prepared November 2026
Full Report · Public edition · Version 1.1

11,365
Global feed-labelled records across the eight OAST, blind-XSS, hook, and recon families

3,744
Endpoints examined by Odin by Brandefense across 2,944 distinct hosts

0
Confirmed blocklist IPv4 addresses published in the Indicators tab (two feed-confirmed German ezXSS rows withheld)

2,481
Dual-use OAST and web reconnaissance endpoints across six tool families (all withheld)

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

How 3,744 OAST, blind-XSS, browser-hook, and web recon endpoints narrow to behavioral callback hunting after filtering 2,481 dual-use tools, 154 PHP 8.5 header collisions, and two single-signal ezXSS rows

01	Executive Summary	04
02	Four Functional Groups, One Analytical Model	07
03	Dual-Use OAST Listeners and Web Reconnaissance Suites	11
04	Feed-Confirmed ezXSS Rows, 154 PHP 8.5 Collisions, and BeEF on Port 3000	14
05	Asset Recon and PentAGI Files on a China Open Directory	18
06	Feed Volume Against Confirmed Infrastructure	21
07	What Defenders Should Do	23
08	Detection Rules	25
09	Indicators	28
10	Scope and Limits	31
11	Sources	33
A	Indicators of compromise (0)	35
B	Detection rules (4 files)	36

IN SHORT

Across a historical feed sample through 20 November 2026, Odin by Brandefense reviewed 3,744 endpoints on 2,944 hosts across eight web offense and recon families. We also reviewed one open directory in China reporting 3,001 files. Zero endpoints qualified for IP blocking after setting aside 2,481 dual-use endpoints, 154 PHP 8.5 header collisions, 800 port-3000 BeEF candidates, and two feed-confirmed German ezXSS rows lacking independent HTTP corroboration. Hunt outbound `/callback` and OAST traffic, and enforce strict admin Content Security Policies.

01

CHAPTER 01

Executive Summary

IN SHORT

Across a historical feed sample through 20 November 2026, Odin by Brandefense reviewed 3,744 endpoints on 2,944 hosts across eight web offense and recon families. We also reviewed one open directory in China reporting 3,001 files. Zero endpoints qualified for IP blocking after setting aside 2,481 dual-use endpoints, 154 PHP 8.5 header collisions, 800 port-3000 BeEF candidates, and two feed-confirmed German ezXSS rows lacking independent HTTP corroboration. Hunt outbound `/callback` and OAST traffic, and enforce strict admin Content Security Policies.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Out-of-band application security testing (`OAST`) listeners, blind cross-site scripting (`XSS`) panels, browser-hooking servers, and web reconnaissance suites create a hard problem for defenders using external attack surface management. Internal security teams and hired penetration testers use PortSwigger Burp Collaborator, DNSLog, reNgine, Burp Suite, ARL-Next, and reconFTW during authorized assessments every day. Attackers use those exact same tools to confirm blind server-side vulnerabilities and map public assets.

Feeding raw threat lists for these eight tool families straight into a firewall blocks authorized scanners, disrupts cloud mail ports, and cuts off ordinary PHP 8.5 websites. We reviewed all 3,744 endpoints and 538 sampled directory entries without running any captured script or contacting any external host.

1. **Dual-use OAST and recon tools call for egress hunting, not IP blocks.**
   Six commercial and open-source testing tools account for 2,481 of the 3,744 examined endpoints on 1,841 hosts, drawn from 10,131 feed-labelled records. Every dual-use address is withheld from blocklists, including 939 Burp Collaborator listeners and 800 reNgine consoles. With cyber threat intelligence, defenders can prioritize outbound DNS or HTTP calls from production application servers to OAST domains instead of blocking dual-use cloud addresses.

2. **Benign PHP 8.5 response headers triggered 154 false ezXSS alerts.**
   Among 463 examined ezXSS blind-XSS records on 304 hosts, 154 suspected endpoints across 123 hosts were ordinary PHP 8.5 websites on AWS, Hetzner, and Contabo. Their standard `X-Powered-By: PHP/8.5` headers matched an overly broad upstream feed signature. Two feed-confirmed German HTTP servers on M247 Europe SRL and DigitalOcean, LLC also stay withheld until independent HTTP or `/callback` responses corroborate them.

3. **One open directory in China held ARL-plus and PentAGI files.**
   Separate from the endpoint feeds, an open HTTP directory in China reports 3,001 exposed files, with 538 entries sampled. The directory held an ARL-plus Docker asset-mapping repository, two PentAGI autonomous penetration-testing trees, Langfuse tracing clients, Fast Reverse Proxy configurations, and JavaScript command-injection proofs of concept. Because the server's owner is not established, its address, port, paths, and file hashes are withheld.

Three Decisions for Leaders

1. **Hunt blind-XSS `/callback` posts and lock down admin browsers.** Inspect egress proxy logs for HTTP `POST` requests to `/callback`, and enforce strict `Content-Security-Policy` `script-src` rules on internal support, ticketing, and administration panels.
2. **Hunt outbound OAST callbacks from production servers instead of blocking dual-use tools.** Keep Burp Collaborator, DNSLog, reNgine, Burp Suite, ARL-Next, and reconFTW feeds out of firewall blocklists. Alert whenever Java, PHP, Node.js, Python, or .NET application servers resolve OAST domains.
3. **Reject banner-only and port-only threat feeds while auditing internal test labs.** Remove rules that flag PHP 8.5 version headers or TCP port 3000 without payload proof, and ban unauthenticated HTTP file servers on red-team virtual machines.

The Study in Numbers

Number	What it is
11,365	Global feed-labelled records across the eight OAST, blind-XSS, hook, and recon families
3,744	Endpoints examined by Odin by Brandefense across 2,944 distinct hosts
0	Confirmed blocklist IPv4 addresses published in Appendix A (two feed-confirmed German ezXSS rows withheld)
2,481	Dual-use OAST and web reconnaissance endpoints across six tool families (all withheld)
154	Benign PHP 8.5 web endpoints on 123 hosts falsely flagged as suspected ezXSS servers
800	Single-signal TCP port 3000 BeEF candidates across 800 hosts (0 confirmed)
3,001 / 538	Reported exposed files and sampled entries on the China open-directory server

How to use this report.

Security leaders can start with Chapters 1, 6, and 7. Application security, WebOps, and platform engineers should read Chapters 3, 4, and 7. SOC analysts and detection engineers should go directly to Chapters 8 and 9. Threat intelligence analysts will find the full cohort breakdown, the China open-directory analysis, and scope limits in Chapters 2, 5, 6, and 10.

02

CHAPTER 02

Four Functional Groups, One Analytical Model

IN SHORT

Zero shared IP addresses or certificates connect the two feed-confirmed ezXSS servers or the China open directory to the dual-use cohorts. We present the four groups together as an analytical model of web reconnaissance and callback operations. Hunt outbound callbacks, and do not infer a shared actor.

INSIDE THIS CHAPTER

How the eight families and the open directory compare 08

Why upstream status labels cannot drive firewall blocks 08

02 Four Functional Groups, One Analytical Model

Web intrusions often combine four building blocks. Reconnaissance suites map the attack surface: subdomains, open ports, and web routes. Out-of-band callback listeners confirm blind server bugs that return no error in the HTTP response.

Blind-XSS and browser-hook panels capture administrator sessions when stored payloads render inside internal tools. Autonomous offense stacks then chain reconnaissance output into automated exploitation attempts.

In raw threat feeds, all four layers arrive mixed together under broad `confirmed` or `suspected` tags. Treating every feed entry as hostile breaks authorized penetration tests, blocks cloud mail ports, and blacklists legitimate web servers.

We compared host addresses and TLS certificates across all 2,944 hosts in the 3,744-endpoint set and the China open directory. Zero shared hosts connect the two feed-confirmed ezXSS servers or the China open directory to any other family. Within the dual-use and candidate cohorts, six hosts expose two tools at once.

Four of those six hosts pair ARL-Next with DNSLog, one pairs reconFTW with reNgine, and one pairs Burp Collaborator with an unconfirmed ezXSS candidate. Because no technical indicator links these populations to a single threat actor, we treat them as an analytical model. Figure 1 shows the four groups and how defenders should handle each one.

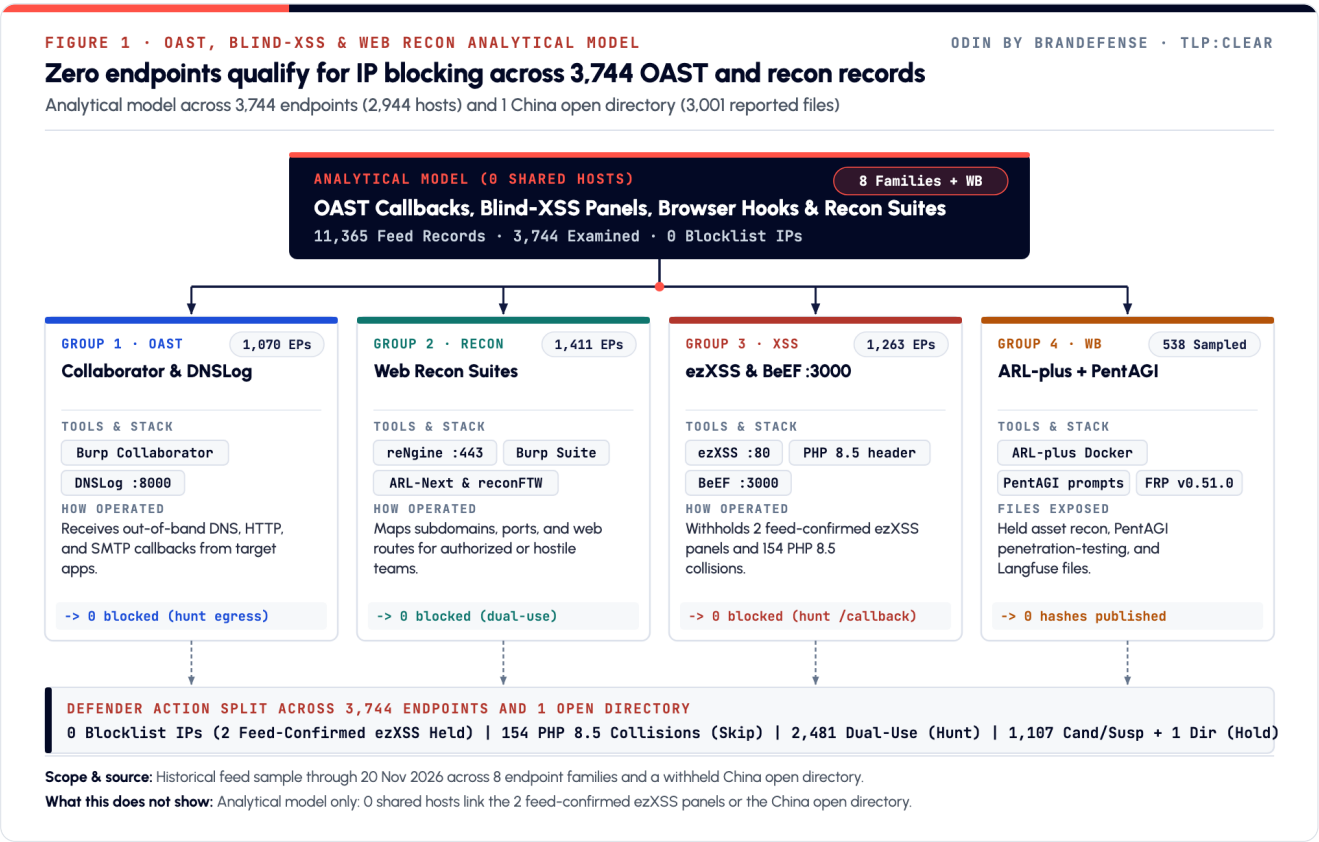


Figure 1: Zero endpoints qualify for IP blocking across 3,744 OAST and recon records

How the eight families and the open directory compare

Across all eight endpoint families, upstream feeds listed 11,365 global records, including 127 records marked verified by upstream providers. Our review examined 3,744 endpoints on 2,944 distinct hosts (2,950 summed per-family hosts before accounting for six hosts present in two families).

Family or workspace	Functional role	Global feed records	Feed verified	Countries	Examined records	Hosts	Confirmed (block)	Filtered PHP 8.5	Withheld (hunt or skip)
Burp Collaborator	Commercial OAST listener	4,295	73	42	939	428	0	0	939
reNgine	Open-source web recon	5,094	23	67	800	727	0	0	800
BeEF	Browser hook framework	810	0	30	800	800	0	0	800
ezXSS	Blind-XSS callback panel	424	2	27	463	304	0	**154**	309
Burp Suite	Commercial web testing suite	306	6	28	306	271	0	0	306
ARL-Next	Asset recon lighthouse	226	20	13	226	217	0	0	226
DNSLog	OAST DNS and HTTP callback	131	3	10	131	124	0	0	131
reconFTW	Automated recon wrapper	79	0	10	79	79	0	0	79
8-Family Subtotal	**Endpoint cohorts**	**11,365**	**127**	**--**	**3,744**	**2,944**	**0**	**154**	**3,590**
ARL-plus and PentAGI	Open directory (China, port withheld)	3,001 files	25 listings	1	538 (220 detailed)	1	0	0	538

The 463 examined ezXSS endpoints exceed the 424-record feed snapshot by 39; the supplied evidence does not establish when those rows went offline.

Why upstream status labels cannot drive firewall blocks

Upstream feed statuses across the 3,744 examined endpoints label 137 endpoints `confirmed`, 274 `suspected`, and 3,333 `candidate`. Automating blocks from those upstream labels would misfire immediately for two reasons.

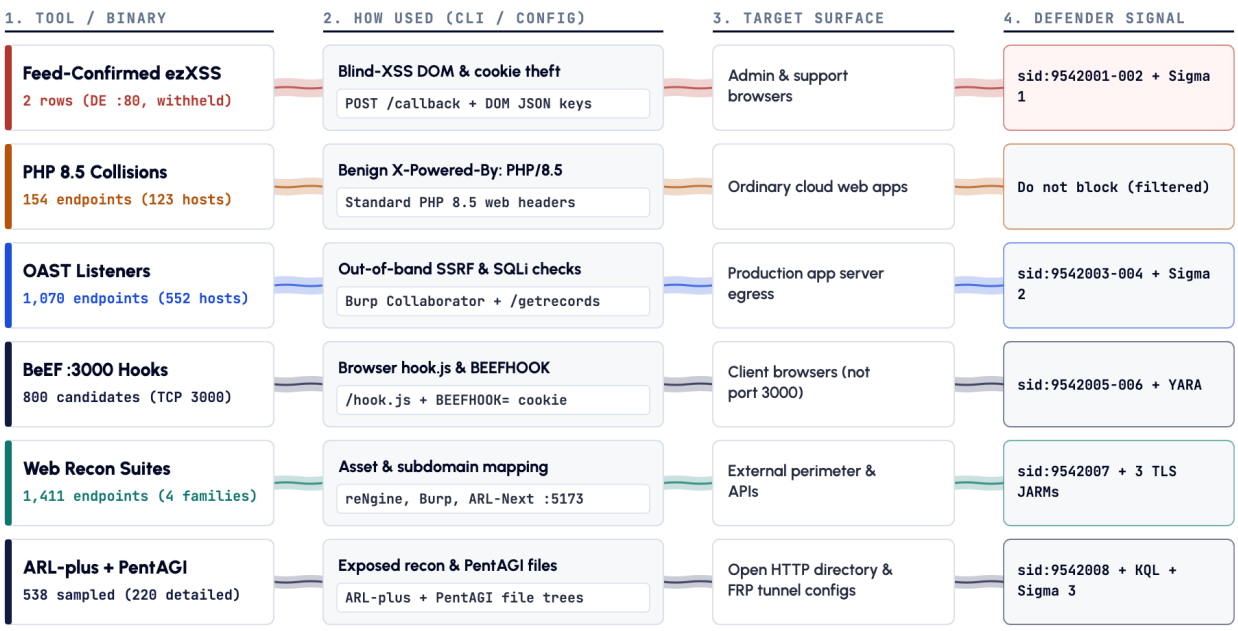
First, 135 of the 137 upstream `confirmed` endpoints belong to dual-use assessment tools, while the remaining two ezXSS rows lack independent HTTP callback proof. Those 135 dual-use rows comprise 83 Burp Collaborator listeners, 23 reNgine consoles, 20 ARL-Next servers, 6 Burp Suite instances, and 3 DNSLog servers.

Second, 154 of the 274 upstream `suspected` endpoints (56.2% overall, and 76.2% of the 202 suspected ezXSS rows) are harmless PHP 8.5 websites. Figure 2 maps each slice of the dataset to the specific network, DNS, and host detections that replace blind IP blocking.

FIGURE 2 · INFRASTRUCTURE TAXONOMY & DETECTION MATRIX
ODIN BY BRANDEFENSE · TLP:CLEAR

Behavioral HTTP and DNS rules separate possible callbacks from PHP 8.5 noise

Six infrastructure slices across 3,744 endpoints and 538 open-directory entries mapped to defender controls



Scope & source: Historical feed sample through 20 Nov 2026 examined by Odin by Brandefense.
What this does not show: All 3,744 endpoint addresses (including 2 feed-confirmed ezXSS rows) and the China open-directory host and port stay withheld.

Figure 2: Behavioral HTTP and DNS rules separate possible callbacks from PHP 8.5 noise

CHAPTER 03

Dual-Use OAST Listeners and Web Reconnaissance Suites

IN SHORT

Six dual-use testing families account for 2,481 examined endpoints across 1,841 hosts, led by Burp Collaborator (939 endpoints) and reNgine (800 endpoints). Because authorized testers share this infrastructure, defenders should hunt outbound callbacks from production servers instead of blocking dual-use addresses.

INSIDE THIS CHAPTER

Self-hosted Burp Collaborator (939 endpoints) and DNSLog (131 endpoints) 11

Four web reconnaissance suites (1,411 endpoints) 12

03 Dual-Use OAST Listeners and Web Reconnaissance Suites

Whether an OAST or web reconnaissance server represents a threat depends on which side of your perimeter starts the connection. Inbound requests from reNgine or Burp Suite look like everyday internet scanning.

An outbound DNS query or HTTP request from an internal Java, PHP, Python, or Node.js application server to Burp Collaborator or DNSLog is completely different. Production servers have no business reason to contact OAST listeners. An outbound hit points to active blind server-side request forgery (`SSRF`), blind SQL injection, XML external entity (`XXE`), or remote code execution testing (T1071.001).

Self-hosted Burp Collaborator (939 endpoints) and DNSLog (131 endpoints)

PortSwigger's Burp Collaborator lets testers run private callback servers that capture DNS lookups, HTTP/HTTPS requests, and SMTP mail interactions on one host. Across 4,295 global feed records in 42 countries, we examined 939 Burp Collaborator endpoints on 428 hosts, an average of 2.19 open ports per host.

The port breakdown reflects Collaborator's multi-protocol listener design:

- **Web callback ports (811 endpoints).** HTTPS port 443 (394 endpoints, 42.0%) and HTTP port 80 (387 endpoints, 41.2%) dominate. Docker high-port mappings on 33443 (15 endpoints) and 3380 (15 endpoints) follow, alongside 8080 (11) and 8443 (10).
- **Mail submission ports (61 endpoints).** Thirty-four endpoints listen on SMTPS port 465 and 27 listen on SMTP submission port 587. Blocking port-based Collaborator feeds without protocol checks risks interfering with mail-submission workflows.
- **Western cloud concentration.** Hosting centers on commercial cloud networks used by enterprise AppSec teams: Amazon.com, Inc. (275 endpoints, 29.3%), DigitalOcean, LLC (181, 19.3%), and OVH SAS (86, 9.2%). Microsoft Corporation (52, 5.5%), Hetzner Online GmbH (39), and Akenes SA (32) follow. The United States (338), Germany (153), the United Kingdom (67), and France (66) lead geographically.
- **Java JSSE TLS fingerprint.** Across TLS-enabled ports, 425 Burp Collaborator endpoints (45.3%) share one JARM fingerprint (`28d28d28d00028d00042d42d00042de0d120da90d5910c5de2af43b74a6bd8`). That JARM comes from Collaborator's embedded Java JSSE TLS stack and appears on zero other families in our 3,744-endpoint set. Another 217 Collaborator endpoints behind Nginx or OpenSSL reverse proxies share a second JARM (`2ad2ad16d2ad2ad00042d42d00042ddb04deffa1705e2edc44cae1ed24a4da`). Direct HTTP responses also carry the `Server: Burp Collaborator` header.

Open-source `DNSLog` servers (131 endpoints on 124 hosts across 10 countries) show a different regional and port profile. Nearly seven in ten sit in China (60) and Hong Kong (29), followed by the United States (22) and Singapore (8). Tencent (38 combined) and Alibaba (30 combined) host the largest shares.

Over three-fifths (79 of 131, 60.3%) expose the default Python/Django `DNSLog-Platform` web interface on TCP port 8000, where users poll `/getrecords` to view captured DNS queries. Four hosts in China and the United States run DNSLog alongside ARL-Next on the same server.

Four web reconnaissance suites (1,411 endpoints)

The four web reconnaissance and vulnerability-scanning families span 1,411 examined endpoints (5,705 global feed records) and split cleanly by default management port and cloud footprint:

Recon tool	Global feed	Examined / hosts	Main TCP ports	Leading countries & providers	Notable TLS or cert trait
reNgine	5,094	800 / 727	443 (690), 8000 (47), 8082 (18), 8443 (18)	US (246), DE (128), FR (70) · Contabo (157), DigitalOcean (133), AWS (46)	205 share exclusive Caddy/ Nginx JARM; 35 use `TRAEFIK DEFAULT CERT`
Burp Suite	306	306 / 271	80 (106), 443 (94), 8080 (53), 8081 (13)	IE (135), US (77), KR (14) · AWS (197, 64.4%), Korea Telecom (6), SK Broadband (6)	37 share one exclusive Burp TLS JARM
ARL-Next	226	226 / 217	5173 (175, 77.4%), 443 (15), 80 (8), 5003 (5)	CN (149), HK (38), US (18) · Tencent (105), Alibaba (33), LUCIDACLOUD (16)	Vite UI on 5173; 24 present `CN=localhost`
reconFTW	79	79 / 79	8001 (78, 98.7%), 8080 (1)	US (27), DE (17), IN (9) · DigitalOcean (45, 57.0%), Contabo (7), Akamai (6)	One German host also runs reNgine on 443 and 8000

All 2,481 endpoints across these six dual-use families remain withheld. Blocking AWS Ireland scanner nodes, DigitalOcean recon droplets, or Korean telecom addresses would create constant operational friction without stopping an attacker from spinning up a new VPS in minutes.

CHAPTER 04

Feed-Confirmed ezXSS Rows, 154 PHP 8.5 Collisions, and BeEF on Port 3000

IN SHORT

Across 463 ezXSS records and 800 BeEF candidates, two feed-confirmed German ezXSS HTTP servers stay withheld pending independent HTTP or callback corroboration. Over three-quarters of suspected ezXSS feed records (154 of 202) were false positives on ordinary PHP 8.5 websites. Hunt outbound `POST /callback` requests and `BEEFHOOK` cookies, and enforce strict admin Content Security Policies.

INSIDE THIS CHAPTER

Two feed-confirmed ezXSS rows in Germany withheld pending corroboration 14

How PHP 8.5 response headers caused 154 false ezXSS alerts 15

Why all 800 BeEF port-3000 candidates are withheld 16

04 Feed-Confirmed ezXSS Rows, 154 PHP 8.5 Collisions, and BeEF on Port 3000

Unlike server-side OAST tools, blind-XSS (`ezXSS`) and browser-hooking (`BeEF`) frameworks target human staff inside back-office web applications (T1059.007). An attacker plants an HTML `<script>` tag inside a support ticket, contact form, `User-Agent` string, or order comment.

An internal administrator or support analyst later views that record inside an unhardened dashboard. The browser then loads the external script and posts session data back to the attacker's listener. Figure 3 contrasts possible blind-XSS and server OAST callbacks against the PHP 8.5 response-header collision.

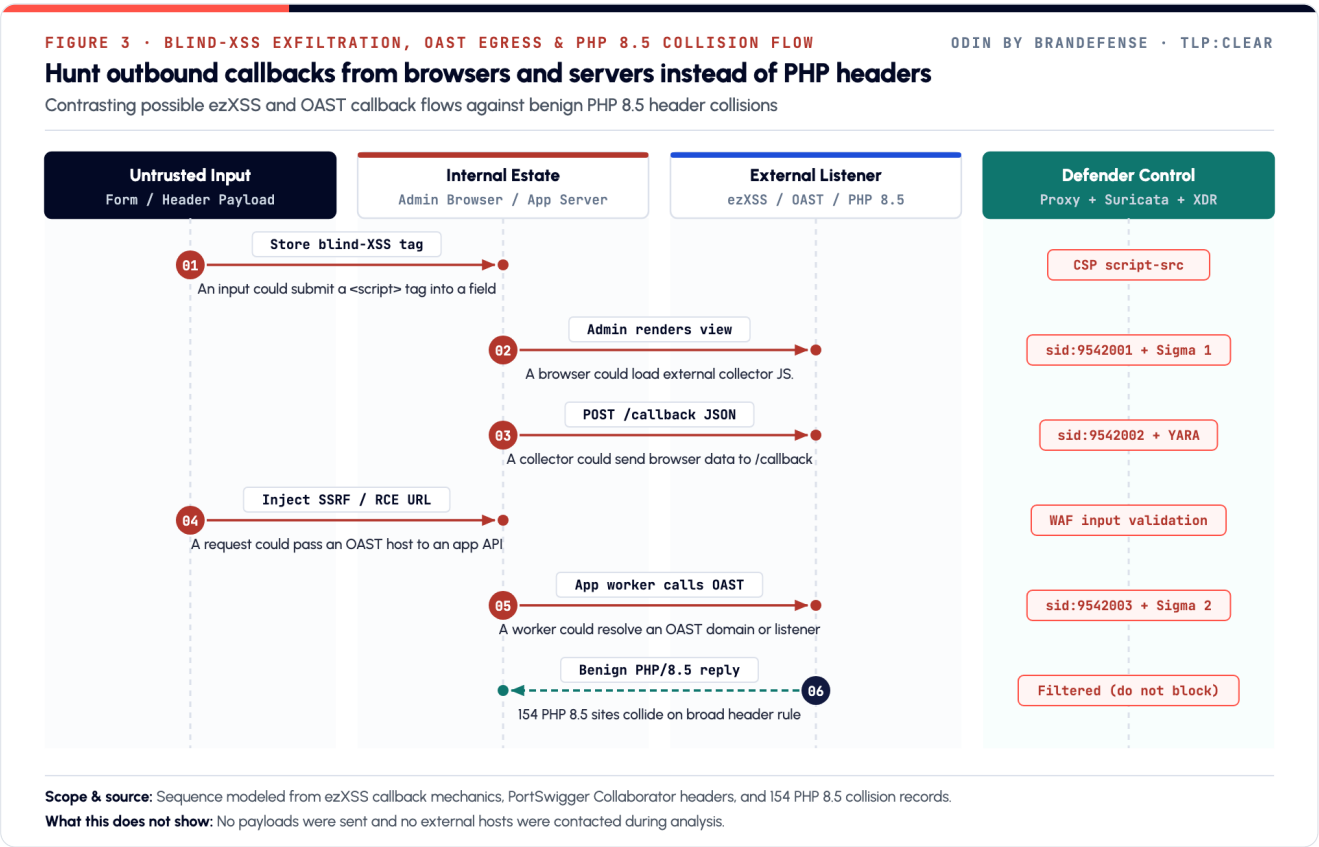


Figure 3: Hunt outbound callbacks from browsers and servers instead of PHP headers

Two feed-confirmed ezXSS rows in Germany withheld pending corroboration

Across 463 examined ezXSS endpoints on 304 hosts in 27 countries, two feed-confirmed rows require independent HTTP or callback corroboration before either address can be published or blocked.

Feed-confirmed row	Location and provider	Protocol	Disposition
German ezXSS row 1 (address withheld)	Germany, M247 Europe SRL	Plain HTTP (:80)	Withhold IP; hunt outbound `POST /callback`
German ezXSS row 2 (address withheld)	Germany, DigitalOcean, LLC	Plain HTTP (:80)	Withhold IP; hunt outbound `POST /callback`

Both servers were observed on 13 November 2026 on TCP port 80. Because the record carries no independent HTTP body or callback capture, both IPv4 addresses and their certificate domains stay withheld.

When an ezXSS collector script runs in a browser, it posts JSON to `/callback`. The payload can contain `uri`, `cookies`, `referer`, `user-agent`, `origin`, `localstorage`, `sessionstorage`, `dom`, and optional `html2canvas` screenshots. Inspecting egress proxy traffic for that JSON callback structure catches blind-XSS exfiltration even when an operator rotates to a new IP address.

How PHP 8.5 response headers caused 154 false ezXSS alerts

Examining the 202 ezXSS endpoints marked `suspected` by upstream feeds uncovered a major false-positive trap. Exactly **154 endpoints across 123 hosts** were benign PHP 8.5 websites rather than blind-XSS panels. They made up 76.2% of suspected ezXSS rows and 33.3% of all 463 examined ezXSS rows.

The `ezXSS` application is written in PHP. When PHP's `expose_php` configuration directive is left enabled (`expose_php = On`), web responses include an `X-Powered-By: PHP/...` header alongside standard `PHPSESSID` cookies and web security headers. As cloud administrators deployed **PHP 8.5** (`X-Powered-By: PHP/8.5.x`), an overly broad upstream feed signature matched that response-header combination. It labelled ordinary websites as `suspected` ezXSS servers without checking for `/callback` or `/manage` routes.

The hosting profile of those 154 collision endpoints confirms they are mainstream cloud web applications:

- **Cloud and port distribution.** Ninety-eight of the 154 collision endpoints (63.6%) sit on Amazon.com, Inc., followed by Hetzner Online GmbH (17), Contabo GmbH (13), Alibaba (7), and Contabo Inc. (4). Ninety listen on HTTP port 80, 46 on HTTPS port 443, and 8 on port 8080. Leading countries are the United States (57), Germany (33), South Korea (14), India (11), Australia (9), and China (8).
- **Generic web TLS stacks.** Their top TLS JARM fingerprints (`29d29d00029d29d00041d41d00041dcb3f8752a2d70d1dbf446fc46ab5df96` on 11 endpoints and `27d40d40d00040d00042d43d000000d2e61cae37a985f75ecafb81b33ca523` on 10 endpoints) are standard cloud reverse-proxy fingerprints shared across unrelated web services.

Pushing upstream `suspected` ezXSS records into a web proxy blocklist would have cut off 123 legitimate PHP 8.5 websites. All 154 PHP 8.5 collision endpoints are suppressed. The remaining 309 ezXSS records also stay withheld. They include 2 feed-confirmed German HTTP rows, 48 suspected rows on 37 hosts, 251 candidates on 147 hosts, 4 Magyar Telekom ISP rows, and 4 shared-certificate collisions.

Why all 800 BeEF port-3000 candidates are withheld

We examined 800 `BeEF` (Browser Exploitation Framework) endpoints across 800 hosts in 30 countries, drawn from 810 global feed records with 0 feed-verified rows. Every examined BeEF endpoint sits on **TCP port 3000** with `candidate` status:

Slice	Endpoints	Hosts	Ports	Key hosting context	Action
ezXSS feed-confirmed HTTP rows	2	2	80 (2)	Germany: M247 Europe SRL (1), DigitalOcean (1)	Withhold IP; hunt `/callback`
ezXSS PHP 8.5 header collision	154	123	80 (90), 443 (46), 8080 (8)	AWS (98), Hetzner (17), Contabo (13)	Do not block
ezXSS suspected and candidates	307	184	443 (155), 80 (111), 8443 (8)	48 suspected, 251 candidate, 4 Magyar Telekom ISP, 4 shared cert	Withhold IP; hunt `/callback`
BeEF port-3000 candidates	779	779	3000 (779)	CN (352), US (154), DE (69) · Tencent (168), Alibaba (141), AWS (114)	Withhold IP; hunt `BEEFHOOK`
BeEF telecom ISP and shared cert	21	21	3000 (21)	14 telecom ISP (China Telecom, China Unicom, Free SAS) + 7 shared cert	Withhold IP

BeEF listens on TCP port 3000 by default (`/hook.js` and `/ui/authentication`). However, TCP 3000 is also the default port for Grafana dashboards, Metabase, Gitea, Node.js, Next.js, and Ruby on Rails servers. Certificate subjects on the 800 port-3000 hosts include `TRAEFIK DEFAULT CERT` (8), `localhost` (5), and `CloudFlare Origin Certificate` (4). Instead of blocking port-3000 addresses, defenders should detect BeEF by its `BEEFHOOK` and `beef.init` strings in HTTP responses and the `BEEFHOOK=` session cookie in outbound requests.

CHAPTER 05

Asset Recon and PentAGI Files on a China Open Directory

IN SHORT

An open HTTP directory in China reports 3,001 exposed files, with 538 sampled and 220 exported in detail. It held an ARL-plus Docker repository, two PentAGI autonomous penetration-testing directory trees, Langfuse tracing code, and Fast Reverse Proxy tunnel configurations. Audit red-team test servers for unauthenticated HTTP directories and unmanaged reverse-proxy tunnels.

INSIDE THIS CHAPTER

Seven file groups inside the 220 detailed entries 19

What the ARL-plus, PentAGI, Langfuse, and FRP file groups contain 19

05 Asset Recon and PentAGI Files on a China Open Directory

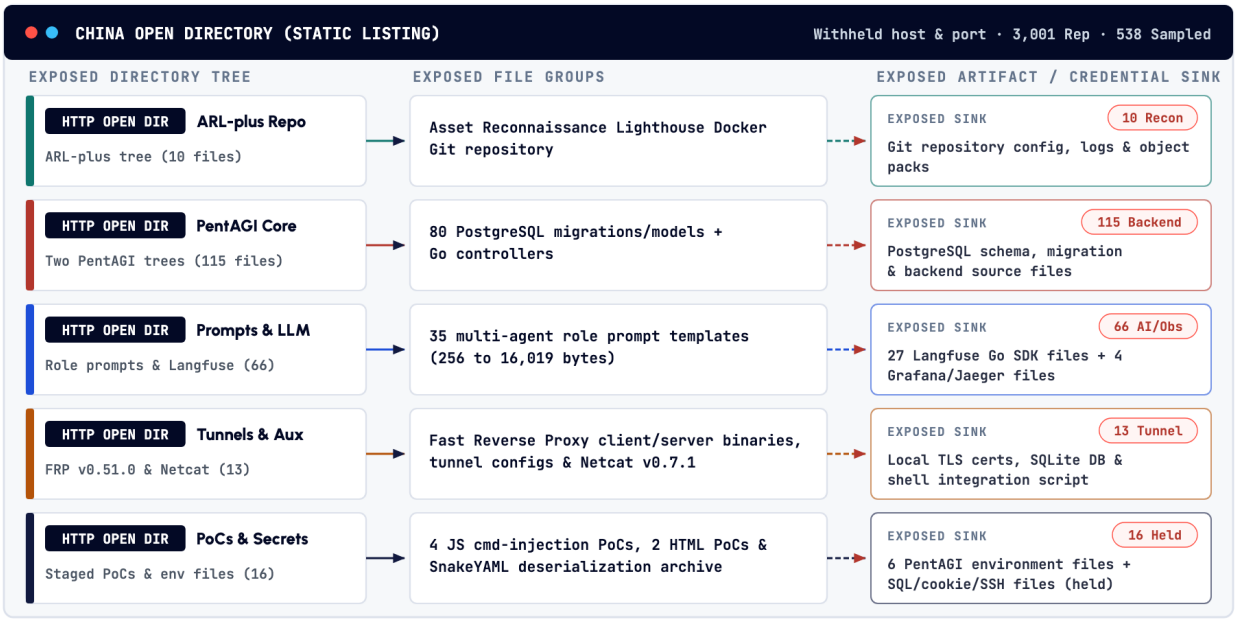
Endpoint feeds show individual ARL-Next or reNgine login pages from the outside. Separately, an open HTTP directory observed in China held files from several security tools. That server held files associated with asset reconnaissance and autonomous penetration testing; the evidence does not establish an operator or show that the tools ran.

Across 25 directory listings, the server reported 3,001 exposed files, with 538 entries sampled through 20 November 2026 and 220 entries exported in detail. Eighteen detailed repository entries carry last-seen dates of 16 August 2026, while the remaining detailed entries carry last-seen dates from 18 to 20 November 2026. Figure 4 breaks down the 220 detailed entries across five file groups.

FIGURE 4 · OPEN-DIRECTORY FILE GROUP BREAKDOWN — ODIN BY BRANDEFENSE · TLP:CLEAR

One China open directory holds ARL-plus recon and PentAGI file groups

Structural breakdown of 220 detailed inventory entries (from 538 sampled / 3,001 reported files)



Scope & source: Historical sample through 20 Nov 2026 from a withheld China open directory (25 site records).
What this does not show: Static listing does not establish an operator or show that the tools ran; host address, port, paths, and hashes are withheld.

Figure 4: One China open directory holds ARL-plus recon and PentAGI file groups

Seven file groups inside the 220 detailed entries

Across the 220 detailed entries, there are 81 SQL files, 42 Go source files, 35 prompt templates, and 21 extensionless files. Another 25 entries include 7 HTML pages, 4 JavaScript files, 4 Markdown documents, 4 configuration files, and 6 Git pack files. The remaining 16 entries comprise 12 archive, certificate, or database files and 4 other text files:

File group in open directory	Detailed entries	Share	Published hashes	What the directory exposes
PentAGI backend, SQL, and Git source	115	52.3%	0	80 PostgreSQL migrations and models across two PentAGI directory trees, Go controllers, a 285 KB Go documentation file, and 5 search HTML fixtures
PentAGI agent prompt templates	35	15.9%	0	Multi-agent role prompt templates ranging from 256 to 16,019 bytes
Langfuse, Grafana, and Jaeger tracing	31	14.1%	0	27 Langfuse Go SDK client files plus 4 Grafana and Jaeger observability files
Tunneling, proxy, and shell tools	13	5.9%	0	Fast Reverse Proxy v0.51.0 client and server binaries and configuration files, Netcat v0.7.1, a shell integration script, and local TLS and SQLite files
ARL-plus Docker recon repository	10	4.5%	0	ARL-plus Docker Git repository configuration, branch logs, and object packs
Environment, database, cookie, and SSH files	9	4.1%	0	Six PentAGI environment configuration files, one SQL dump, one cookie text file, and an SSH host-key backup
Exploit PoCs and deserialization archive	7	3.2%	0	Four JavaScript command-injection PoC scripts, two HTML PoC pages, and a SnakeYAML Java deserialization archive

What the ARL-plus, PentAGI, Langfuse, and FRP file groups contain

The directory held files from three categories of security and tunneling software, without showing whether they were run together:

- **External discovery (`ARL-plus`).** Community `ARL-plus` Docker stacks automate subdomain enumeration, port scanning, service fingerprinting, and vulnerability template checks.
- **Autonomous penetration testing (`PentAGI` and `Langfuse`).** Two PentAGI directory trees (59 and 128 entries) hold Go backend and PostgreSQL schema files. They also expose 35 agent role prompt templates, with the five largest measuring 16,019, 12,544, 11,676, 11,133, and 10,486 bytes. Twenty-seven Langfuse client files provide LLM observability support.
- **Tunnels and manual PoCs (`FRP` and `SnakeYAML`).** Fast Reverse Proxy v0.51.0 binaries and configuration files sit beside the repository trees. The directory also holds a SnakeYAML archive for Java deserialization testing and four JavaScript command-injection PoCs.

Starting an unauthenticated HTTP server over that workspace exposed six environment configuration files, local TLS private keys, a SQLite database, and cookie files at once. An authorized red team or security researcher can make that exact configuration mistake on a cloud test machine. We therefore withhold the server's address, port, directory paths, file names, and file hashes.

CHAPTER 06

Feed Volume Against Confirmed Infrastructure

IN SHORT

Zero of the 3,744 examined endpoints qualify for an IP blocklist once single-signal feed rows are withheld. Keeping dual-use OAST and recon tools, 154 PHP 8.5 header collisions, and 800 port-3000 candidates off blocklists prevents self-inflicted outages. Replace static IP feeds with outbound `/callback`, `BEEFHOOK`, and OAST domain hunting.

06 Feed Volume Against Confirmed Infrastructure

A threat feed is only as useful as the false positives it removes. Figure 5 compares the global feed and examined volume of each family against the zero confirmed blocklist IPv4 addresses that remained after multi-signal filtering.

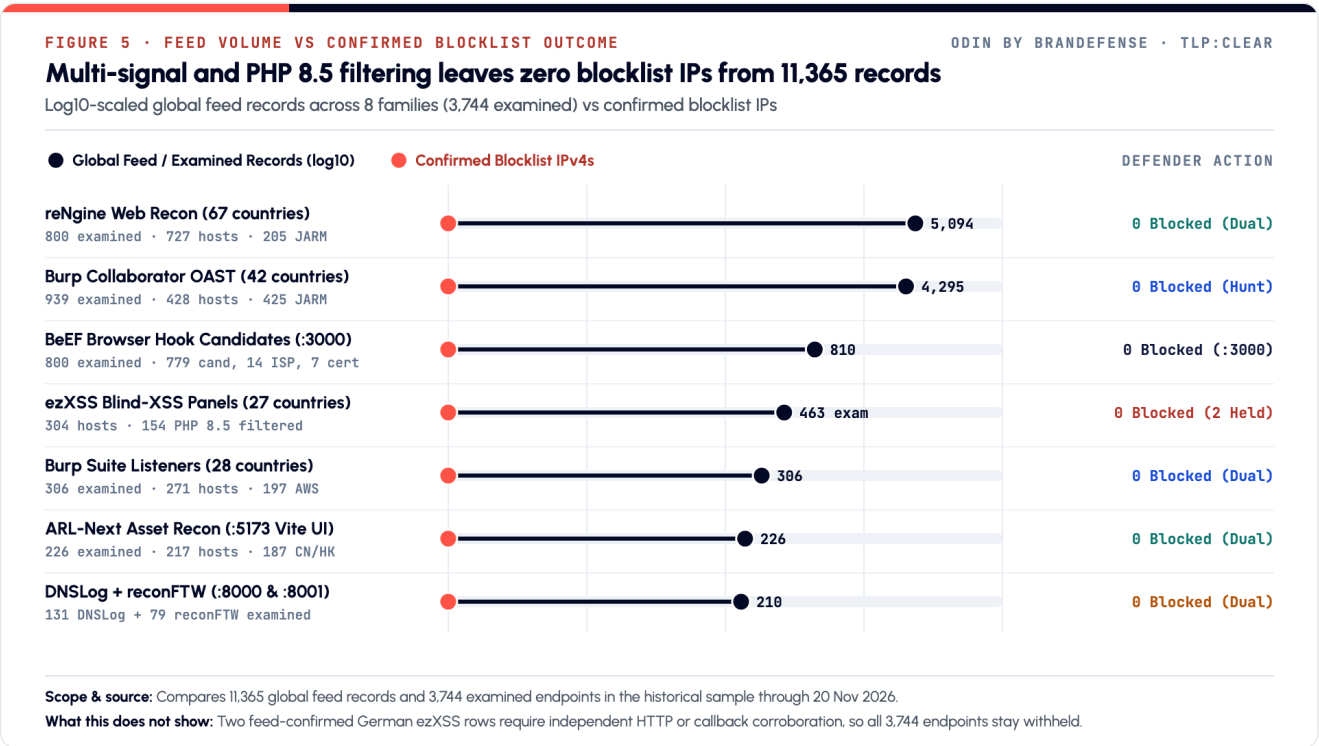


Figure 5: Multi-signal and PHP 8.5 filtering leaves zero blocklist IPs from 11,365 records

Four plain analytical boundaries govern how defenders should read these findings:

1. **Separate populations, not one actor.** Zero shared IP addresses, certificates, or domains connect the two feed-confirmed German ezXSS servers, the China open directory, and the dual-use cohorts. Grouping them is an analytical model for defenders, not an attribution claim.
2. **PHP 8.5 headers are not blind-XSS proof.** Over three-quarters of upstream `suspected` ezXSS hits (154 of 202) were ordinary PHP 8.5 websites on AWS, Hetzner, and Contabo. Never block an `ezXSS` feed entry unless payload or `/callback` behavior confirms it.
3. **Dual-use scanners and shared JARMs must stay hunt-only.** All 2,481 endpoints across Burp Collaborator, reNgine, Burp Suite, ARL-Next, DNSLog, and reconFTW are dual-use security tools. Generic reverse-proxy JARMs also cross families: one Go/proxy JARM (`27d40d40d00040d00042d43d000000d2e61cae37a985f75ecafb81b33ca523`) appears on 62 endpoints across six families, including 10 PHP 8.5 collision sites. Traefik's default JARM (`3fd3fd00000000000043d3fd3fd43d79451d8c63b099acafdbabb24551d0e6`) appears on 56 endpoints across seven families. Even the three single-family JARMs for Burp Collaborator, reNgine, and Burp Suite identify software stacks rather than malicious intent, so they are strictly for hunting.
4. **Staged workbench files do not confirm target breaches.** Nothing in the China open-directory listing shows whether `ARL-plus`, `PentAGI`, the SnakeYAML deserialization archive, or the JavaScript PoC scripts were used against external victims or inside an authorized lab.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

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

07 What Defenders Should Do

Network and web proxy

- [ ] Withhold single-signal ezXSS feed addresses from firewall blocklists until independent HTTP response or callback evidence corroborates them.
- [ ] Search the last 30 days of web proxy logs for any internal workstation that sent HTTP `POST` requests to `/callback` containing `localstorage`, `sessionstorage`, and `cookies`.

Internal admin panels and web applications

- [ ] Deploy a strict `Content-Security-Policy` (`script-src 'self'` or cryptographic nonces, plus tight `connect-src` rules) and `HttpOnly` session cookies across internal dashboards and support portals. That prevents stored blind-XSS payloads from loading external scripts or reading session cookies.
- [ ] Set `expose_php = Off` in production `php.ini` files to stop emitting `X-Powered-By: PHP/8.5.x` headers, and strip header-only `ezXSS` or port-only :3000 `BeEF` feeds from automated firewall blocklists.

Production server egress

- [ ] Restrict default outbound internet DNS and HTTP access from production application server tiers (`Java`, `PHP-FPM`, `Node.js`, `Python`, `.NET`), and alert whenever a server worker resolves `.oastify[.]com`, `.burpcollaborator[.]net`, `.dnslog[.]cn`, or `.ceye[.]io`.

Red-team and security testing infrastructure

- [ ] Audit internal security-testing virtual machines for `Directory listing for /` pages and unmanaged `frpc` reverse-proxy tunnels. Ban `python3 -m http.server` bound to `0.0.0.0` in folders holding environment configuration or Langfuse keys, and store agent API keys in short-lived vaults.

CHAPTER 08

Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

How the Suricata signatures work 25

How the endpoint, DNS, and file rules work 26

MITRE ATT&CK mapping 26

08 Detection Rules

The full rules are in Appendix B. All four published rule files pass static syntax checks; they were not loaded on live Suricata, Sentinel, Sigma, or YARA engines on our build host. Test and tune them in your own environment before enabling inline blocking or automated isolation.

Rule file	What it catches
`odin_burp_collaborator_ezxss_beef_and_arl_recon.rules`	Suricata 7: eight signatures covering ezXSS collector script delivery, `/callback` JSON exfiltration, `Server: Burp Collaborator` headers, DNSLog `/getrecords` polling, BeEF `hook.js` and `BEEFHOOK=` traffic, ARL `/api/task` requests, and Python `SimpleHTTP` open-directory listings
`sentinel_oast_ezxss_beef_and_recon_hunt.kql`	Microsoft Sentinel and Defender XDR: URL-only hunt for outbound `/callback` connections on port 80, application-server processes contacting OAST domains or ports 3380/33443, and unapproved `http.server` or `frpc` process launches
`sigma_oast_callback_ezxss_and_recon_workbench.yml`	Sigma (3 rules): web proxy outbound HTTP `POST` requests to ezXSS `/callback` endpoints, Linux application workers resolving OAST callback domains, and Linux `http.server` or `frpc` execution outside approved lab hosts
`odin_ezxss_beef_and_pentagi_arl_artifacts.yar`	YARA (4 rules): ezXSS DOM collector payloads, BeEF `hook.js` client scripts, PentAGI and Langfuse configuration files, and captured Python `SimpleHTTP` open-directory listings or FRP tunnel configuration files

How the Suricata signatures work

SID	Looks for	Why
9542001	Inbound HTTP 200 response body containing `/callback`, `localstorage`, and `sessionstorage`	Catches ezXSS blind-XSS collector script delivery to internal browsers
9542002	Outbound `POST` to `/callback` carrying `localstorage`, `sessionstorage`, and `cookies`	Catches ezXSS blind-XSS DOM and cookie exfiltration to any external host
9542003	Inbound HTTP response containing `Server: Burp Collaborator`	Hunt signal for internal hosts interacting with a self-hosted Burp Collaborator server
9542004	Outbound `GET` to `/getrecords` with a `Cookie:` header	Hunt signal for DNSLog callback polling from internal hosts
9542005	Inbound HTTP 200 response body containing `BEEFHOOK` and `beef.init`	Catches BeEF browser-hook JavaScript delivery regardless of server port
9542006	Outbound HTTP request carrying `Cookie: BEEFHOOK=`	Catches command polling from a browser already hooked by BeEF
9542007	Inbound `POST` to `/api/task` with a `Token:` header and `target` field	Hunt signal for unplanned ARL or ARL-Next scan task submissions
9542008	HTTP 200 response carrying `Server: SimpleHTTP/` and `Directory listing for /`	Flags exposed Python `http.server` open-directory listings during perimeter or lab scans

Keep SIDs 9542003, 9542004, and 9542007 in alert-only mode where internal AppSec or red-team scanners operate.

How the endpoint, DNS, and file rules work

The Sentinel KQL query looks back seven days across `DeviceNetworkEvents` and `DeviceProcessEvents`. Because `DeviceNetworkEvents.RemoteUrl` records only a URL or domain, the first branch runs a URL-only hunt (`HUNT_POSSIBLE_CALLBACK_URL`). It matches outbound TCP port 80 connections whose `RemoteUrl` contains `/callback`, with no host exclusions. Confirming HTTP `POST` methods and JSON body keys (`localstorage`, `sessionstorage`, and `cookies`) requires web proxy telemetry, covered by Suricata SID 9542002 and Sigma Rule 1.

Second, it hunts `java`, `php-fpm`, `node`, `python3`, `dotnet`, `w3wp.exe`, `nginx`, `httpd`, or `apache2` processes reaching OAST domains (`oastify[.]com`, `burpcollaborator[.]net`, `dnslog[.]cn`, `ceye[.]io`). That same query branch also matches `/getrecords`, `BEEFHOOK`, or Collaborator container ports 3380 and 33443. Third, it flags `python3 -m http.server` or `frpc` / `frps` runs (`server_addr`, `remote_port`, `vhost_http_port`) outside exact hostnames in `ApprovedAppSecHosts` and `ApprovedLabHosts`.

In Sigma, Rule 1 (`high`) alerts on web proxy outbound HTTP `POST` requests to `/callback` carrying `localstorage`, `sessionstorage`, and `cookies` with no exclusions. Rule 2 (`high`) watches Linux DNS logs for `java`, `php-fpm`, `node`, `python3`, `dotnet`, or `ruby` resolving `.oastify[.]com`, `.burpcollaborator[.]net`, `.dnslog[.]cn`, or `.ceye[.]io` outside named AppSec scanner hosts. Rule 3 (`high`) flags Linux `http.server` (`http.server`, `SimpleHTTPServer`) or `frpc` / `frps` (`server_addr`, `remote_port`) processes outside a named lab host.

YARA rule	Fires when
`ODIN_ezXSS_Blind_XSS_Collector_Payload`	A file under 256 KB combines `/callback`, `localstorage`, and `sessionstorage` with `document.cookie`, `html2canvas`, or `ezXSS`
`ODIN_BeEF_Browser_Hook_Script`	A script under 512 KB combines `BEEFHOOK` and `beef.init` with `beef[.]net` or `beef.browser`
`ODIN_PentAGI_Autonomous_Offense_Prompt_Bundle`	A file under 128 KB combines `PENTAGI` with Langfuse observability configuration keys (`LANGFUSE_SECRET_KEY`, `LANGFUSE_PUBLIC_KEY`, or `LANGFUSE_BASE_URL`)
`ODIN_Exposed_ARL_PentAGI_OpenDir_Listing`	A file under 512 KB matches Python `http.server` `Directory listing for /` HTML tags or an FRP tunnel configuration block (`[common]`, `server_addr`, and `remote_port`)

MITRE ATT&CK mapping

Tactic	Technique	How it appears in this study	Detection coverage
Execution	T1059.007 Command and Scripting Interpreter: JavaScript	ezXSS blind-XSS collectors and BeEF `hook.js` running in browsers	SIDs 9542001, 9542002, 9542005, 9542006; YARA rules 1 and 2
Command and Control	T1071.001 Application Layer Protocol: Web Protocols	Blind-XSS `/callback` posts and application-server OAST callbacks	SIDs 9542002, 9542003, 9542004; KQL; Sigma rules 1 and 2
Reconnaissance	T1595.002 Active Scanning: Vulnerability Scanning	reNgine, Burp Suite, ARL-Next (:5173/ `/api/task`), and reconFTW (:8001)	SID 9542007; 3 hunt-only TLS JARM fingerprints
Resource Development	T1583.003 Acquire Infrastructure: Virtual Private Server	China open directory holding ARL-plus, PentAGI, and Fast Reverse Proxy v0.51.0 files	SID 9542008; KQL; Sigma rule 3; YARA rules 3 and 4
Credential Access	T1552.001 Unsecured Credentials: Credentials In Files	Six PentAGI environment configuration files, Langfuse keys, local TLS keys, and cookies exposed over HTTP	SID 9542008; Sigma rule 3; YARA rules 3 and 4

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Dual-use TLS JARM fingerprints to hunt (report only) 28

Behavioral HTTP and workspace patterns (report only) 29

09 Indicators

Appendix A holds 0 published indicators: 0 IPv4 addresses and 0 file hashes. The two feed-confirmed German ezXSS rows from the register stay withheld until independent HTTP response or callback evidence corroborates them. Eleven hunting patterns appear in this report only: three dual-use TLS JARM fingerprints, six HTTP and cookie patterns, and two lab-exposure patterns. All 3,744 endpoint addresses, the China open-directory server address and port, and all workbench file digests are withheld.

Type	Count	Where to use it
Confirmed blocklist IPv4 addresses	0	Zero endpoint IPv4 addresses are published in Appendix A; two feed-confirmed German ezXSS rows stay withheld pending independent HTTP or callback corroboration
File hashes (SHA-256)	0	Zero workbench file hashes are published because host ownership is not established
Dual-use TLS JARM fingerprints	3	Listed below for TLS hunting only; never block an address on JARM alone
HTTP, cookie, and lab-exposure patterns	8	Listed below for proxy, IDS, and internal lab-hygiene hunting

Dual-use TLS JARM fingerprints to hunt (report only)

JARM fingerprint	Tool stack	Observed count and hunting note
`28d28d28d00028d00042d42d00042de0d120da90d5910c5de2af43b74a6bd8`	Burp Collaborator	425 endpoints (exclusive Java JSSE stack); hunt outbound server connections only
`2ad2ad0002ad2ad0002ad2ad2ad2ada5fc7ec76b7ed0b839211317d3844486`	reNgine	205 endpoints (default Caddy/Nginx TLS stack); hunt only, never block
`29d29d00029d29d00029d29d29d29da8f16e8f0a21d89b06fef5080cbd1b2a`	Burp Suite	37 endpoints (exclusive Burp TLS stack); hunt only, never block

Behavioral HTTP and workspace patterns (report only)

Pattern	Family	Where to hunt
`Server: Burp Collaborator`	Burp Collaborator	Egress HTTP response headers from production server subnets (SID 9542003)
`POST /callback` with `cookies`, `localstorage`, `sessionstorage`	ezXSS	Corporate browser web proxy logs and egress IDS (SIDs 9542001 and 9542002)
`/hook.js` (`BEEFHOOK` + `beef.init`) and `Cookie: BEEFHOOK=`	BeEF	Browser web proxy logs and egress IDS (SIDs 9542005 and 9542006)
`/ui/authentication` on TCP 3000	BeEF	External attack-surface audits separating BeEF from Grafana or Node.js
`GET /getrecords`	DNSLog	Egress HTTP proxy logs (SID 9542004)
`POST /api/task/` with `Token:` header	ARL / ARL-Next	Internal and perimeter web logs (SID 9542007)
`Server: SimpleHTTP/` with `Directory listing for /`	Python `http.server`	Open-directory scans and red-team host audits (SID 9542008)
`python3 -m http.server` or `frpc` with `server_addr`	Lab exposure and FRP	Endpoint process audits on security-testing hosts (KQL and Sigma rule 3)

10

CHAPTER 10

Scope and Limits

IN SHORT

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

10 Scope and Limits

This report covers 3,744 endpoints drawn from 11,365 global feed records through 20 November 2026. Of those endpoints, 2,081 were last seen before 7 November 2026. It also covers one open HTTP directory in China reporting 3,001 files, with 538 sampled and 220 detailed entries observed in August and November 2026.

Our analysis is strictly passive and static. We did not run any script from the open directory, replay any exposed environment credential, or send traffic to any external target or callback domain.

Nothing in the dataset confirms whether any downstream organization was breached through the two feed-confirmed ezXSS servers, the other endpoints, or the China open directory. Because dual-use scanners, single-signal feed rows, PHP 8.5 collision sites, and the China open-directory server lack corroborated malicious ownership, all addresses stay withheld.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. **PortSwigger Documentation**, Burp Collaborator Overview and Private Collaborator Server Architecture: portswigger.net/burp/documentation/collaborator (verified on 21 November 2026).
2. **PortSwigger Web Security Academy**, Blind SSRF Vulnerabilities and Out-of-Band (OAST) Detection Mechanics: portswigger.net/web-security/ssrf/blind (verified on 21 November 2026).
3. **OWASP Foundation**, Cross Site Scripting (XSS) and Stored/Blind XSS Attack Mechanics: owasp.org/www-community/attacks/xss/ (verified on 21 November 2026).
4. **OWASP Cheat Sheet Series**, Cross Site Scripting Prevention Cheat Sheet and Content Security Policy Defenses: cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (verified on 21 November 2026).
5. **PHP Official Manual**, Description of Core php.ini Directives (`expose_php` and `X-Powered-By` Response Header): www.php.net/manual/en/ini.core.php (verified on 21 November 2026).
6. **MITRE ATT&CK**, Enterprise Sub-Technique T1071.001: Application Layer Protocol - Web Protocols: attack.mitre.org/techniques/T1071/001/ (verified on 21 November 2026).
7. **MITRE ATT&CK**, Enterprise Sub-Technique T1059.007: Command and Scripting Interpreter - JavaScript: attack.mitre.org/techniques/T1059/007/ (verified on 21 November 2026).
8. **MITRE ATT&CK**, Enterprise Sub-Technique T1552.001: Unsecured Credentials - Credentials In Files: attack.mitre.org/techniques/T1552/001/ (verified on 21 November 2026).

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

No individual indicators are published for this study; Appendix A explains why. Appendix B carries every published detection rule in full.

INSIDE

A. Indicators of compromise (0) 35

B. Detection rules (4 files) 36

A Indicators of Compromise

No individual addresses or file hashes 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 in Appendix B.

B Detection Rules

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

odin_burp_collaborator_ezxss_beef_and_arl_recon.rules

SURICATA

```
# Odin by Brandefense - Monograph 42 / 46 Suricata 7.x Detection Suite
# Coverage: Behavioral ezXSS blind-XSS collector script delivery and DOM/cookie exfiltration
#           callbacks, outbound server callbacks to PortSwigger Burp Collaborator and DNSLog
#           listeners, BeEF hook.js delivery and BEEFHOOK session polling, ARL / ARL-Next
#           task submission APIs, and Python SimpleHTTP open-directory listings
# Classification: TLP:CLEAR

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R42] ezXSS Blind-XSS Collector Script Delivery";
flow:established,from_server; http.stat_code; content:"200"; http.response_body; bsize:>64; content:"/callback";
fast_pattern; content:"localstorage"; content:"sessionstorage"; classtype:web-application-attack; sid:9542001;
rev:1; metadata:created_at 2026_11_21, deployment perimeter_egress, tactic execution;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R42] ezXSS Blind-XSS Browser DOM and Cookie
Exfiltration Callback POST"; flow:established,to_server; http.method; content:"POST"; http.uri; bsize:>8;
content:"/callback"; fast_pattern; http.request_body; bsize:>64; content:"localstorage";
content:"sessionstorage"; content:"cookies"; classtype:web-application-attack; sid:9542002; rev:1;
metadata:created_at 2026_11_21, deployment perimeter_egress, tactic collection;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R42] HUNT PortSwigger Burp Collaborator Out-of-Band
HTTP Server Header"; flow:established,from_server; http.header; bsize:>24; content:"Server: Burp Collaborator";
fast_pattern; classtype:attempted-recon; sid:9542003; rev:1; metadata:created_at 2026_11_21, deployment
perimeter_egress, policy behavioral_hunt;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R42] HUNT DNSLog Out-of-Band Callback Polling Request
(/getrecords)"; flow:established,to_server; http.method; content:"GET"; http.uri; bsize:>10; content:"/
getrecords"; fast_pattern; http.header; content:"Cookie:"; classtype:attempted-recon; sid:9542004; rev:1;
metadata:created_at 2026_11_21, deployment perimeter_egress, policy behavioral_hunt;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R42] BeEF Browser Exploitation Framework Hook Script
Delivery"; flow:established,from_server; http.stat_code; content:"200"; http.response_body; bsize:>48;
content:"BEEFHOOK"; fast_pattern; content:"beef.init"; classtype:trojan-activity; sid:9542005; rev:1;
metadata:created_at 2026_11_21, deployment perimeter_egress, tactic execution;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R42] BeEF Hooked Browser Session Cookie Polling
(BEEFHOOK)"; flow:established,to_server; http.cookie; bsize:>16; content:"BEEFHOOK="; fast_pattern;
classtype:trojan-activity; sid:9542006; rev:1; metadata:created_at 2026_11_21, deployment perimeter_egress,
tactic command_and_control;)

alert http any any -> $HOME_NET any (msg:"ODIN [R42] HUNT Asset Reconnaissance Lighthouse (ARL / ARL-Next) Task
API Submission"; flow:established,to_server; http.method; content:"POST"; http.uri; bsize:>8; content:"/api/
task"; fast_pattern; http.header; content:"Token:"; http.request_body; bsize:>12; content:"target";
classtype:attempted-recon; sid:9542007; rev:1; metadata:created_at 2026_11_21, deployment internal_and_perimeter,
policy behavioral_hunt;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R42] Python SimpleHTTP Open-Directory Listing
Response"; flow:established,from_server; http.stat_code; content:"200"; http.header; content:"Server:
SimpleHTTP/"; http.response_body; bsize:>32; content:"Directory listing for /"; fast_pattern;
classtype:attempted-recon; sid:9542008; rev:1; metadata:created_at 2026_11_21, deployment perimeter, tactic
resource_development;)
```

odin_ezxss_beef_and_pentagi_arl_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Monograph 42 / 46 YARA Artifact Suite
   Scope: Structural detection rules for ezXSS blind-XSS DOM collector payloads,
          BeEF (Browser Exploitation Framework) hook.js scripts, PentAGI and Langfuse
          environment configurations, and exposed Python SimpleHTTP open-directory
          listings or Fast Reverse Proxy tunnel configuration files.
   Classification: TLP:CLEAR
*/

rule ODIN_ezXSS_Blind_XSS_Collector_Payload
{
    meta:
        author = "Odin by Brandefense"
        monograph = "42 / 46"
        classification = "TLP:CLEAR"
        action = "BLOCK_BLIND_XSS_COLLECTOR"
        description = "Matches ezXSS blind-XSS JavaScript DOM/cookie collector scripts and callback templates
without triggering on generic PHP 8.5 headers"
    strings:
        $cb1 = "/callback" ascii
        $k1 = "localstorage" ascii nocase
        $k2 = "sessionstorage" ascii nocase
        $k3 = "document.cookie" ascii
        $k4 = "html2canvas" ascii nocase
        $k5 = "ezXSS" ascii nocase
    condition:
        filesize < 256KB and $cb1 and $k1 and $k2 and ($k3 or $k4 or $k5)
}

rule ODIN_BeEF_Browser_Hook_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "42 / 46"
        classification = "TLP:CLEAR"
        action = "BLOCK_BEEF_BROWSER_HOOK"
        description = "Matches BeEF (Browser Exploitation Framework) hook.js client scripts by session cookie and
command-polling primitives"
    strings:
        $b1 = "BEEFHOOK" ascii
        $b2 = "beef.init" ascii
        $b3 = "beef.net" ascii
        $b4 = "beef.browser" ascii
    condition:
        filesize < 512KB and $b1 and $b2 and ($b3 or $b4)
}

rule ODIN_PentAGI_Autonomous_Offense_Prompt_Bundle
{
    meta:
        author = "Odin by Brandefense"
        monograph = "42 / 46"
        classification = "TLP:CLEAR"
        action = "HUNT_AUTONOMOUS_PENTEST_WORKSPACE"
        description = "Matches PentAGI penetration-testing configuration files pairing PentAGI settings with
Langfuse observability keys"
    strings:
        $p1 = "PENTAGI" ascii nocase
        $l1 = "LANGFUSE_SECRET_KEY" ascii
        $l2 = "LANGFUSE_PUBLIC_KEY" ascii
        $l3 = "LANGFUSE_BASE_URL" ascii
    condition:
        filesize < 128KB and $p1 and 1 of ($l*)
}

rule ODIN_Exposed_ARL_PentAGI_OpenDir_Listing
{
    meta:
```

```
        author = "Odin by Brandefense"
        monograph = "42 / 46"
        classification = "TLP:CLEAR"
        action = "CONTAIN_EXPOSED_RECON_WORKBENCH"
        description = "Matches captured Python SimpleHTTP open-directory HTML listings or Fast Reverse Proxy
tunnel configuration files"
    strings:
        $dir1 = "<title>Directory listing for /" ascii
        $dir2 = "<h1>Directory listing for /" ascii
        $frp1 = "[common]" ascii
        $frp2 = "server_addr" ascii
        $frp3 = "remote_port" ascii
    condition:
        filesize < 512KB and (($dir1 and $dir2) or ($frp1 and $frp2 and $frp3))
}
```

sentinel_oast_ezxss_beef_and_recon_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 42 / 46 Microsoft Sentinel & Defender XDR Hunt
// Hunts possible outbound /callback URLs on TCP port 80 (URL-only signal; use web proxy logs for HTTP POST and
JSON body checks),
// application-server outbound OAST callbacks to Burp Collaborator or DNSLog listeners,
// and unapproved ad-hoc Python http.server directory listings or FRP tunnels outside approved lab hosts
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedAppSecHosts = dynamic(["APPSEC-SCANNER-01", "REDTEAM-COLLAB-01"]);
let ApprovedLabHosts = dynamic(["REDTEAM-LAB-LNX-01"]);
let EzxssCallbackHits =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where RemotePort == 80 and RemoteUrl contains "/callback"
    | project EventTime = Timestamp, DeviceId, DeviceName, InitiatingProcessAccountName,
              InitiatingProcessFileName, InitiatingProcessCommandLine, RemoteIP, RemotePort,
              RemoteUrl, SignalClass = "HUNT_POSSIBLE_CALLBACK_URL";
let ServerOastCallbacks =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAppSecHosts) or DeviceName in~ (ApprovedLabHosts))
    | where InitiatingProcessFileName in~ ("java", "java.exe", "php-fpm", "php-cgi", "node", "python3", "dotnet",
"w3wp.exe", "nginx", "httpd", "apache2")
    | where RemoteUrl has_any ("oastify.com", "burpcollaborator.net", "dnslog.cn", "ceye.io", "/getrecords",
"BEEFHOOK")
        or RemotePort in (3380, 33443)
    | project EventTime = Timestamp, DeviceId, DeviceName, InitiatingProcessAccountName,
              InitiatingProcessFileName, InitiatingProcessCommandLine, RemoteIP, RemotePort,
              RemoteUrl, SignalClass = "HUNT_SERVER_OAST_CALLBACK";
let WorkbenchProcSignals =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAppSecHosts) or DeviceName in~ (ApprovedLabHosts))
    | where ((
        FileName in~ ("python", "python3") and ProcessCommandLine has "http.server"
    )
    or (
        FileName in~ ("frpc", "frps") and ProcessCommandLine has_any ("server_addr", "remote_port",
"vhost_http_port")
    ))
    | project EventTime = Timestamp, DeviceId, DeviceName,
              InitiatingProcessAccountName = AccountName,
              InitiatingProcessFileName = FileName,
              InitiatingProcessCommandLine = ProcessCommandLine,
              RemoteIP = "", RemotePort = 0, RemoteUrl = "",
              SignalClass = "CONTAIN_EXPOSED_RECON_AGENT_WORKBENCH";
union EzxssCallbackHits, ServerOastCallbacks, WorkbenchProcSignals
| sort by EventTime desc
```

sigma_oast_callback_ezxss_and_recon_workbench.yml

SIGMA

```
title: Outbound Web Proxy HTTP POST to ezXSS Blind-XSS Callback Endpoint
id: 6e18b4c2-3f71-4a90-8d52-202611214201
status: experimental
description: Detects outbound HTTP POST requests from internal workstations or browsers to ezXSS blind-XSS /
callback endpoints carrying DOM and session storage keys. No host exclusions apply so compromised admin browsers
or test hosts cannot mask a blind-XSS callback.
author: Odin by Brandefense
date: 2026/11/21
tags:
  - attack.command-and-control
  - attack.t1071.001
  - attack.execution
  - attack.t1059.007
logsource:
  category: proxy
detection:
  selection_ezxss_callback:
    cs-method: 'POST'
    c-uri|endswith: '/callback'
    cs-body|contains|all:
      - 'localstorage'
      - 'sessionstorage'
      - 'cookies'
  condition: selection_ezxss_callback
falsepositives:
  - Authorized blind-XSS validation tests against isolated staging portals
level: high
---
title: Application Server Worker Querying Out-of-Band OAST Callback Domains
id: 9a42d7f1-8c05-4e36-b129-202611214202
status: experimental
description: Detects production web or application server runtimes (Java, PHP-FPM, Node.js, Python, .NET)
resolving public or self-hosted OAST callback domains (oastify.com, burpcollaborator.net, dnslog.cn, ceye.io),
indicating blind SSRF, SQLi, XXE, or template injection validation.
author: Odin by Brandefense
date: 2026/11/21
tags:
  - attack.initial-access
  - attack.t1190
  - attack.command-and-control
  - attack.t1071.001
logsource:
  category: dns_query
  product: linux
detection:
  selection_worker:
    Image|endswith:
      - '/java'
      - '/php-fpm'
      - '/php-cgi'
      - '/node'
      - '/python3'
      - '/dotnet'
      - '/ruby'
  selection_oast_domain:
    QueryName|endswith:
      - '.oastify.com'
      - '.burpcollaborator.net'
      - '.dnslog.cn'
      - '.ceye.io'
  filter_appsec_hosts:
    ComputerName:
      - 'APPSEC-SCANNER-01'
      - 'REDTEAM-COLLAB-01'
  condition: selection_worker and selection_oast_domain and not filter_appsec_hosts
falsepositives:
  - Scheduled dynamic application security testing runs targeting non-allowlisted staging servers
level: high
```

```
---
title: Ad-Hoc Python HTTP Directory Server or FRP Tunnel on Linux Host
id: 3d75a9e4-2b18-4c63-9f07-202611214203
status: experimental
description: Detects Linux execution of Python http.server directory listings or Fast Reverse Proxy (frpc/frps)
tunnel clients and servers outside approved red-team lab hosts.
author: Odin by Brandefense
date: 2026/11/21
tags:
  - attack.resource-development
  - attack.t1583.003
  - attack.credential-access
  - attack.t1552.001
logsource:
  category: process_creation
  product: linux
detection:
  selection_http_or_frp:
    Image|endswith:
      - '/python'
      - '/python3'
      - '/frpc'
      - '/frps'
  selection_service_args:
    CommandLine|contains:
      - 'http.server'
      - 'SimpleHTTPServer'
      - 'server_addr'
      - 'remote_port'
  filter_lab_hosts:
    ComputerName:
      - 'REDTEAM-LAB-LNX-01'
  condition: selection_http_or_frp and selection_service_args and not filter_lab_hosts
falsepositives:
  - Authorized red-team infrastructure hosts not yet listed in REDTEAM-LAB-LNX-01
level: high
```

ODIN

About Odin Research

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

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

Online edition, indicators and rules · brandefense.io/odin-research/reports/commercial-oast-listeners-blind-xss-panels-and-web-recon-suites/
Short Read · brandefense.io/odin-research/blog/separating-ezxss-callbacks-from-dual-use-oast-scanners-and-php/
Defense Kit · brandefense.io/odin-research/defense-kit/