ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 17 / 30 · CALLBACK INFRASTRUCTURE

Out-of-Band Callback Fabrics

Interactsh, Burp Collaborator, DNSLog and a Cloud Cluster

How self-hosted callback servers turn blind server bugs into signals, and when they sit next to attacker tooling

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

1,794
OAST endpoints on 1,052 IPv4 hosts, after 356 AWS load-balancer reflection collisions were set aside

757
Confirmed or suspected callback nodes (755 listeners plus 2 co-hosted offensive workbenches)

638
Interactsh endpoints among them (523 confirmed, 112 suspected, 3 on dynamic ISPs)

109 and 8
Private Burp Collaborator endpoints (109) and DNSLog listeners (8)

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

How self-hosted callback servers turn blind server bugs into signals, and when they sit next to attacker tooling

01	Executive Summary	04
02	The Callback Fleet and the Four-Role Model	07
03	How the Three Platforms Look on the Wire	11
04	From Blind Probe to Callback: What Exposed Directories Held	14
05	Four Services in Alibaba Cloud 8.148.0.0/16	17
06	Telling Testers from Attackers	20
07	What Defenders Should Do	23
08	Detection Rules	25
09	Indicators	29
10	Scope and Limits	32
11	Sources	34
A	Indicators of compromise (9)	36
B	Detection rules (4 files)	37

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense mapped 1,794 out-of-band callback endpoints on 1,052 hosts. Of these, 757 are confirmed or suspected Interactsh, Burp Collaborator, DNSLog and workbench nodes. Most are dual-use testing servers. But 18 hosts share an IP with other offensive tools, and four offensive services appeared in one Alibaba Cloud subnet on the same day. Block direct DNS, HTTP and SMTP egress from internal servers and route their lookups through Protective DNS.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense mapped 1,794 out-of-band callback endpoints on 1,052 hosts. Of these, 757 are confirmed or suspected Interactsh, Burp Collaborator, DNSLog and workbench nodes. Most are dual-use testing servers. But 18 hosts share an IP with other offensive tools, and four offensive services appeared in one Alibaba Cloud subnet on the same day. Block direct DNS, HTTP and SMTP egress from internal servers and route their lookups through Protective DNS.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Many serious server bugs give no visible answer. A blind server-side request forgery (SSRF) or a blind SQL injection runs quietly in the back end. To confirm such a bug, testers and attackers both use Out-of-Band Application Security Testing (OAST) servers. The probe makes the target server call out to a domain the tester controls, by DNS, HTTPS or mail. A callback shows the server can be reached and can reach the internet.

This report maps self-hosted OAST servers, shows where they share hosts with command-and-control (C2) tools, and walks through one tight cluster of offensive services in Alibaba Cloud.

1. **Most callback servers are dual-use, but 18 share hosts with offensive tools.**
   We count 638 Interactsh, 109 private Burp Collaborator and 8 DNSLog nodes, plus 2 co-hosted offensive workbenches. Most look like ordinary testing servers. Across the fleet, 40 hosts expose four or more listener ports or extra services. Of these, 18 put a callback listener on the same IP as another offensive framework: 8 corroborated and 10 candidates.

2. **Four unlinked offensive services sat in one Alibaba Cloud subnet.**
   On 18 November 2026, Odin first observed a DNSLog collector, a VShell WebSocket listener, a CyberStrikeAI engine and a HexStrikeAI tool bridge in `8.148.0.0/16`. All four sit on separate IPs. Nothing links the four servers, so treat them as separate hosts in a busy cloud range.

3. **Exposed attacker files tie OAST callbacks to exfiltration scripts and Java exploit templates.**
   Open web directories held 40 callback records. Of these, 18 sit on three confirmed offensive servers, in Python DNS exfiltration scripts and Java exploit templates. Another 15 records were incoming probes logged on a scanned third-party web application, not attacker files.

Three Decisions for Leaders

1. **Restrict outbound DNS, HTTP and SMTP from internal application and database servers.** Back-end web servers, middleware containers and databases should resolve names only through internal Protective DNS forwarders.
2. **Watch recursive DNS for callback-style lookups.** Alert when a server process queries a public callback zone or resolves a 26 to 36 character random-looking label.
3. **Block only callback servers with independent offensive evidence.** Raise priority when a listener shares a host with a C2 framework or an offensive workbench. Send standalone listeners to analyst review, because authorized testers run them too.

The Study in Numbers

Number	What it is
1,794	OAST endpoints on 1,052 IPv4 hosts, after 356 AWS load-balancer reflection collisions were set aside
757	Confirmed or suspected callback nodes (755 listeners plus 2 co-hosted offensive workbenches)
638	Interactsh endpoints among them (523 confirmed, 112 suspected, 3 on dynamic ISPs)
109 and 8	Private Burp Collaborator endpoints (109) and DNSLog listeners (8)
4	Complementary offensive services first observed in `8.148.0.0/16` on 18 November 2026
40 and 18	Multi-service hosts (40) and hosts where a listener shares an IP with another offensive framework (18)
40	Callback records found in exposed web directories, 18 of them on confirmed offensive workbenches
9 of 28	Selected indicators ready to block; the other 19 are watchlist, review or reference entries

How to use this report.

Security leaders can read Chapter 1, Chapter 6 and Chapter 7. Network, cloud and database teams should read Chapters 2 to 5 and use the checklist in Chapter 7. Detection engineers and hunters will find the rule logic in Chapter 8 and the indicators in Chapter 9.

02

CHAPTER 02

The Callback Fleet and the Four-Role Model

IN SHORT

The 1,794 endpoints break down into 757 confirmed or suspected nodes and 1,039 candidates. Most are standalone testing servers, so the useful signal is co-hosting with C2 and reconnaissance tools.

INSIDE THIS CHAPTER

An analytical model of four roles 07
Fleet counts by platform 08
Where the fleet is hosted 08
Co-hosting: the strongest signal 09

02 The Callback Fleet and the Four-Role Model

OAST servers fix a visibility problem. Many server-side flaws return no output in the HTTP response. An OAST server runs its own authoritative DNS name server and a wildcard TLS listener. When the target server looks up or fetches a tagged address, the OAST server logs it. That turns a silent bug into a record the tester can read.

For defenders, this creates a choke point. The callback has to leave the internal server. Egress filtering and DNS logging can see it. External listener data alone does not show whether any target was breached.

An analytical model of four roles

Figure 1 is an analytical model, not an observed operation. It shows how four cloud roles could link an attack from probe to implant.

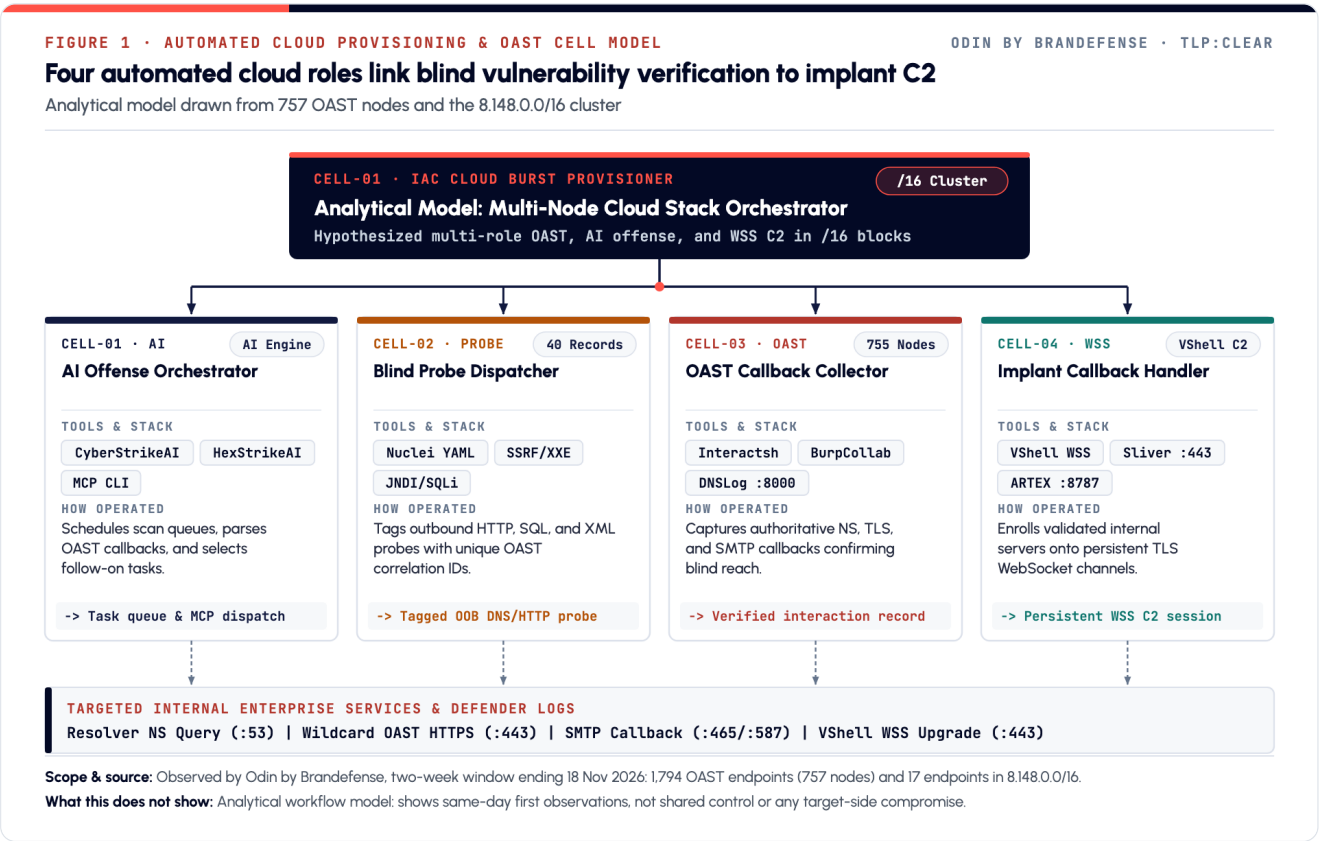


Figure 1: Four automated cloud roles link blind vulnerability verification to implant C2

Role	Example tools	What it does in the model
AI offense orchestrator	CyberStrikeAI, HexStrikeAI, MCP command-line bridges	Queues scans, reads callback results and picks the next task
Blind probe dispatcher	Nuclei YAML templates, SSRF, XXE, JNDI and SQLi payloads	Tags outbound probes with a unique correlation ID
OAST callback collector	Interactsh, Burp Collaborator, DNSLog on 8000	Captures DNS, TLS and SMTP callbacks that confirm blind reach
Implant callback handler	VShell WebSocket, Sliver on 443, ARTEX on 8787	Would hold implant sessions from servers that answered, over a TLS WebSocket channel

MCP, the Model Context Protocol, lets a language model call external tools. Nothing in our data shows one operator running all four roles. The model is a way to read the co-hosting and the subnet cluster in later chapters.

Fleet counts by platform

Each endpoint is an IP and port pair.

Platform	Endpoints (IPs)	Confirmed	Suspected + dynamic ISP	Confirmed or suspected	Candidates
Interactsh	1,245 (700)	523 (6 with open directory)	112 + 3	**638**	607 (6 with open directory)
Burp Collaborator	516 (319)	62	45 + 2	**109**	407
DNSLog	33 (33)	1	7 + 0	**8**	25
Offensive open-directory workbenches	2 (2 shared)	2	0	**2**	0
Total	**1,796 (1,052)**	**588**	**164 + 5**	**757**	**1,039**

The total row counts 1,796 records: the 1,794 OAST endpoints plus the 2 workbenches. A further 356 AWS Application Load Balancer endpoints echo any header they receive. They look like OAST servers but are not, so they are left out of every count.

Where the fleet is hosted

Hosting splits by platform. Interactsh and Burp Collaborator sit mostly on Western cloud providers. DNSLog sits mostly in China.

Platform	Top providers (endpoints)	Top countries (endpoints)
Interactsh	Amazon.com, Inc. (360), DigitalOcean, LLC (180)	United States (575), Germany (145)
Burp Collaborator	Amazon.com, Inc. (142), DigitalOcean, LLC (108)	United States (180), Germany (84)
DNSLog	Tencent (11), Hangzhou Alibaba Advertising Co.,Ltd. (9)	China (20), Hong Kong (6)

Co-hosting: the strongest signal

Across the 1,052 hosts, 40 IPv4 addresses expose several services. Of these, 22 are standalone listener matrices with four or more callback ports. The other 18 put a callback listener on the same IP as a C2, reconnaissance or open-directory service. Eight of those 18 are corroborated and listed below. One of the eight has its address withheld, because its Sliver port is only suspected. The 10 candidate co-locations are withheld.

Host (country)	Hosting network	Callback listener ports	Co-hosted offensive service
Address withheld (US)	NetLab Global	Interactsh 465, 587, 18080	Suspected Sliver on 443, plus an offensive open directory on 8000 (1,417 files)
49.232.168[.]207 (CN)	Shenzhen Tencent	DNSLog 8000 (confirmed)	VShell WebSocket C2 listener on 5030
101.33.231[.]97 (CN)	Shenzhen Tencent	Interactsh 443	Cobalt Strike listener on 8080
43.153.118[.]75 (US)	Shenzhen Tencent	Interactsh 8300	VShell WebSocket listeners on 8084 and 8094
23.94.214[.]197 (US)	HostPapa	Interactsh 465, 587, 443	ARTEX post-exploitation listener on 8787
150.158.9[.]124 (CN)	Shenzhen Tencent	DNSLog 8000	ARL (Asset Reconnaissance Lighthouse) on 5003
51.15.224[.]47 (FR)	Scaleway SAS	Interactsh 80, 443, 465, 587, 3000	ezXSS blind XSS payload manager on 9443
49.51.187[.]59 (US)	Shenzhen Tencent	Interactsh 80 (confirmed)	Offensive open-directory workbench on 8090 (67 files)

CHAPTER 03

How the Three Platforms Look on the Wire

IN SHORT

Interactsh, Burp Collaborator and DNSLog each leave their own headers, routes and TLS fingerprints. Multi-port listeners on 53, 80, 443, 465, 587 and 8000 catch a callback over whichever protocol the firewall leaves open.

INSIDE THIS CHAPTER

Self-hosted Interactsh 11

Private Burp Collaborator 12

DNSLog 12

03 How the Three Platforms Look on the Wire

Public shared callback zones such as `oast[.]pro` and `oastify[.]com` are well known. Many Protective DNS services already sinkhole them. To avoid those blocks, operators can run private collectors on their own domains and cheap cloud servers. Authorized red teams do the same, for the same reason.

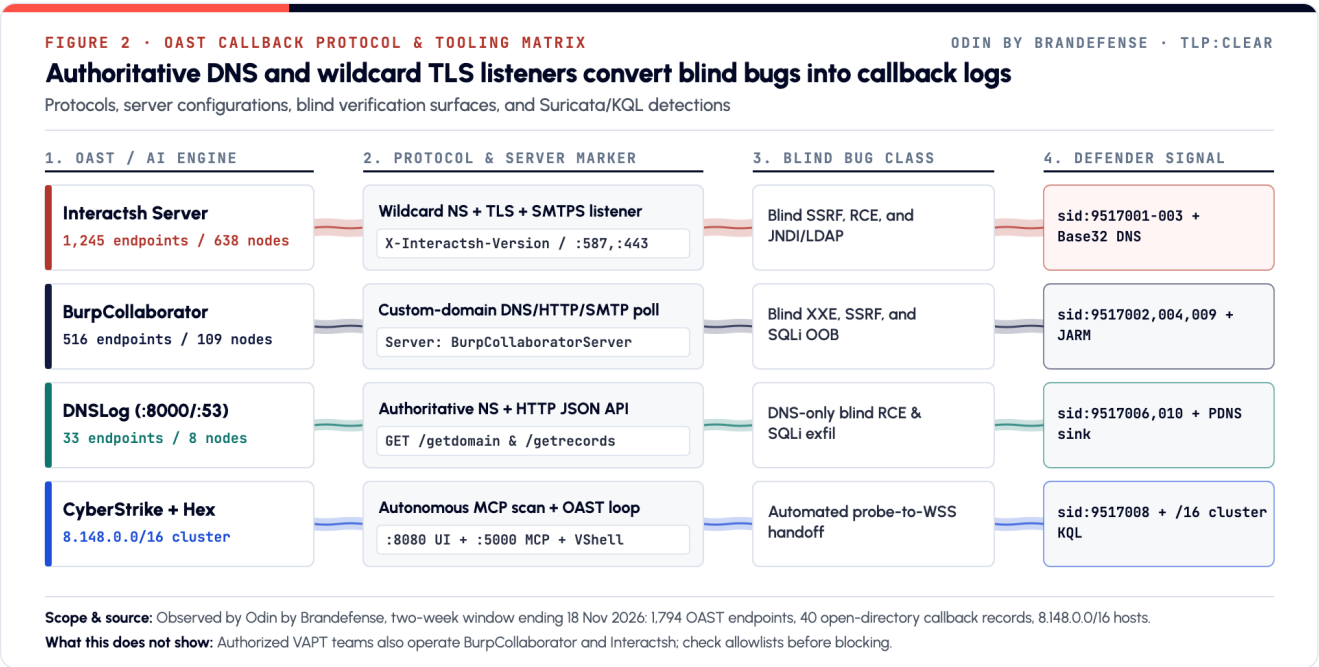


Figure 2: Authoritative DNS and wildcard TLS listeners convert blind bugs into callback logs

Self-hosted Interactsh

ProjectDiscovery's `interactsh-server` runs DNS, HTTP, HTTPS and SMTP collectors under one wildcard domain. It answers with an `X-Interactsh-Version` header or `Server: Interactsh-Server`.

Feature	Value
Endpoints (IPs)	1,245 (700)
Most common ports	80 (374), 587 (303), 443 (267), 465 (216)
TLS fingerprint	229 of 267 HTTPS nodes on 443 (85.8%) share the Go TLS JARM `29d29d00029d29d00041d...`
Client polling route	`/poll?id=` with a `secret` parameter

A JARM value is a fingerprint of how a server answers TLS handshakes. A shared JARM points to the same software stack, not the same owner. Standalone Interactsh matrices with four ports and no other tooling stay on the watchlist.

Private Burp Collaborator

PortSwigger lets customers run a private Collaborator server. It is set up with a JSON file (`serverDomain`, `eventCapture`, `polling`) and answers with `Server: BurpCollaboratorServer`.

Feature	Value
Endpoints (IPs)	516 (319)
Most common ports	443 (213), 80 (199), 465 (34), 587 (27), and 8443, 3380, 33443 (6 each)
TLS fingerprint	236 of 259 TLS nodes (91.1%) share the Java JARM `28d28d28d00028d00042d...`

Several standalone Collaborator servers bind custom certificate domains, and one hub listens on 18 ports. None shows co-hosted C2 tooling, so they stay on the watchlist and their addresses are withheld.

DNSLog

DNSLog pairs an authoritative DNS listener on 53 with an HTTP API. The API runs on 8000 on 18 of 33 endpoints (54.5%).

Two route families exist. `dnslog[.]cn`-compatible panels expose `/getdomain` and `/getrecords` (or the `.php` forms). BugScanTeam's DNSLog uses `/api/` routes instead. Suricata SID 9517006 covers only the first family.

Two DNSLog hosts share an IP with offensive tools: 49.232.168[.]207 with VShell on 5030, and 150.158.9[.]124 with ARL on 5003.

04

CHAPTER 04

From Blind Probe to Callback: What Exposed Directories Held

IN SHORT

Exposed web directories held 40 callback records. Eighteen sit on three confirmed offensive servers, one is a standalone client log, six are template mirrors, and fifteen were logged on a scanned third-party target.

INSIDE THIS CHAPTER

Four blind bug classes that call out 15

The 40 callback records 15

04 From Blind Probe to Callback: What Exposed Directories Held

In manual testing, an analyst pastes one Collaborator address into a request and checks for callbacks by hand. In automated work, scanners such as Nuclei, HexStrikeAI and CyberStrikeAI poll the OAST API themselves. Every probe carries a unique correlation ID, so each callback maps back to one probe.

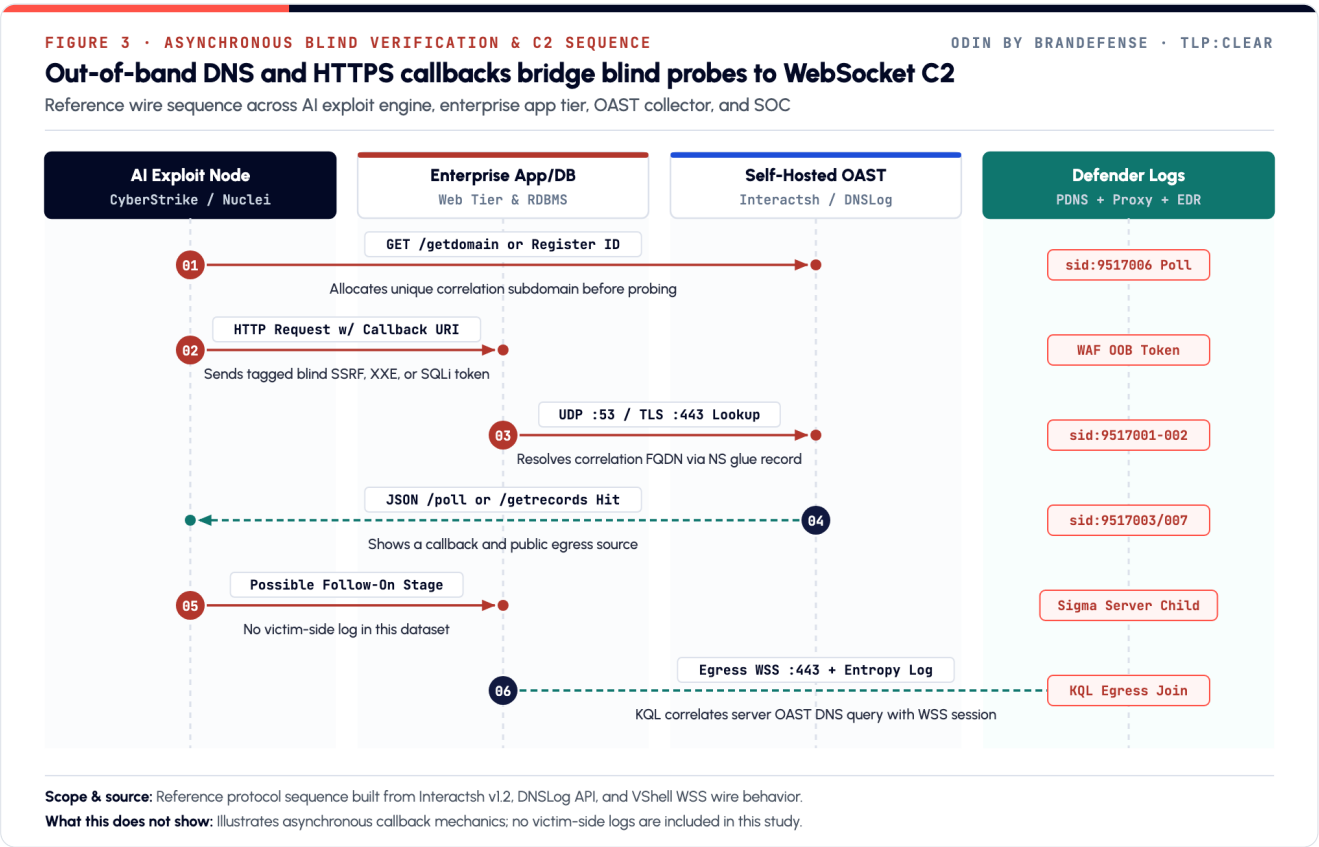


Figure 3: Out-of-band DNS and HTTPS callbacks bridge blind probes to WebSocket C2

Figure 3 is a reference sequence built from public protocol behaviour. Steps 5 and 6 show a possible follow-on stage. No victim-side logs are part of this study.

Four blind bug classes that call out

Firewalls often block inbound traffic but allow recursive DNS on UDP 53 or outbound TLS on 443. That lets a blind bug call out by one of four routes.

Bug class	How the callback happens
Blind SSRF	A web app, webhook previewer or PDF generator fetches an attacker-supplied `https://<correlation-id>.<oast-zone>` URL. This reveals the server's egress IP and HTTP client.
Blind SQL injection	With errors hidden, a query calls database network functions such as `xp_dirtree` (SQL Server) or `UTL_HTTP` and `UTL_INADDR` (Oracle). These trigger a DNS lookup.
Blind XXE and JNDI	An XML parser fetches an external DTD, or a Java logging sink makes an anonymous LDAPv3 bind on 389 or 1389.
Asynchronous command execution	A worker runs `nslookup`, `curl` or `wget` on a multi-label name such as `<hex-chunk>.<correlation-id>.oast.pro`. This confirms execution or leaks short strings.

The 40 callback records

The 40 records use 15 distinct callback names or wildcard patterns across 8 public shared OAST zones. The 18 records on confirmed workbenches use 8 distinct callback names.

Source	Files	Records (distinct)	What they show
144.172.99[.]199:8080, confirmed workbench	`/dns_exfil.py`, `/dns_mass_exfil.py`, `/assemble_dnslog.py`, `/dbg*.log`	12 (5)	DNS exfiltration scripts using `oast[.]pro`, `oast[.]site`, `oast[.]me` and `v9gcgt.dnslog[.]cn`
47.101.51[.]235:8888, confirmed workbench	`/EvilTemplate.java`, `/D2.java`, `/ET.java`, `/M.java`, `/M2.java`	5 (2)	Java exploit templates calling `57v1we22.dnslog[.]cn` and `25cot090vm.dnslog[.]cn`
45.145.228[.]16:8080, confirmed workbench	`/memshell/ExploitScriptEngineFactory.java`	1 (1)	Java memory-shell template calling `wiiesroglrl1t9t7ljwhkjvo1f7cv1.oastify[.]com`
Standalone host (withheld)	`/att_interactsh.out`	1 (1)	Captured Interactsh client output for an `oast[.]pro` name; watchlist
Template mirrors on 5 hosts (withheld)	Nuclei templates and vulnerability wiki pages	6 (5)	Public scanner templates using `*.oast[.]online`, `*.oast[.]pro`, `*.interact[.]sh` and `*.burpcollaborator[.]net`
Scanned third-party web application (withheld)	A Laravel session file	15 (1)	Incoming scanner probes using an `oast[.]site` name, logged by the target

These are staged files. They show intent and tooling. They do not show that any probe or script worked against a real target.

The last row matters for hygiene. That host is a probable target of scanning, not an attacker. It is withheld and should not be blocked.

05

CHAPTER 05

Four Services in Alibaba Cloud 8.148.0.0/16

IN SHORT

On 18 November 2026, Odin first observed four complementary offensive services in Alibaba Cloud `8.148.0.0/16`. They sit inside a wider set of 17 endpoints on 13 hosts in the same range.

INSIDE THIS CHAPTER

The four services in order 17

What links them 17

The rest of the /16 18

05 Four Services in Alibaba Cloud 8.148.0.0/16

Single-host indicators miss how cloud infrastructure clusters. All four services sat in one Alibaba Cloud range (Hangzhou Alibaba Advertising Co.,Ltd., CN).



Figure 4: Four complementary offensive services were first observed in 8.148.0.0/16 on the same day

The four services in order

Port	Service	Framework label	Fingerprint and role
8000	DNSLog	Suspected	HTTP `/getdomain` and `/getrecords` callback collector
443	VShell	Suspected	Go TLS JARM `29d29d16d...`, WebSocket C2 listener; certificate domain withheld
8080	CyberStrikeAI	Confirmed	`CN=CyberStrikeAI`, JARM `40d40d40d...`, autonomous offense engine
5000	HexStrikeAI	Suspected	REST MCP bridge that hands command-line scanners to a model

The four IPs fall between `8.148.195.0/24` and `8.148.233.0/24`. Each exposes a different port and a different role: callback collection, WebSocket C2, AI orchestration and MCP tool execution. Their addresses are withheld.

What links them

Nothing does. We found no shared certificate, cross-host control link or provisioning file on any of the four servers. Alibaba Cloud ranges host many unrelated customers, including testers. The roles fit together on paper, but that is not evidence of one operator.

Because nothing proves shared control, all four hosts stay on the watchlist. They are not on the block list.

The rest of the /16

Beyond the four, the range holds 13 more records on 9 sibling IPs. That makes 17 endpoints on 13 IPs, after setting aside 2 unrelated 3306 false positives.

Only one host in the whole range has the same-host evidence we need for blocking. 8.148.31[.]135 runs a confirmed Metasploit Meterpreter listener on 4446 and 443, plus a Sliver listener on 8888. Its certificate domains are withheld.

The 8 other siblings carry 10 endpoints, all withheld:

Service	Port	Note
CyberStrikeAI	8080	`CN=CyberStrikeAI`; carries an earlier first-seen date of 2 November 2026
PentAGI	8443	Single listener
DNSLog (TLS)	8443	Standalone
ARTEX	8787	Candidate
Hashtopolis (password-cracking manager)	8443	Candidate
Sliver	8443	Candidate
VShell	8090	Candidate
mitmproxy	18081, 18082, 18084	One host, three ports

CHAPTER 06

Telling Testers from Attackers

IN SHORT

Interactsh and Burp Collaborator are everyday tools for contracted testers and bug-bounty researchers. Blocking every listener breaks legitimate work, so only 9 of 28 selected indicators are block-ready.

INSIDE THIS CHAPTER

What earns a block 20

What stays off the block list 21

06 Telling Testers from Attackers

Figure 5 shows how far the confirmed and suspected set shrinks from the full endpoint count for each platform.

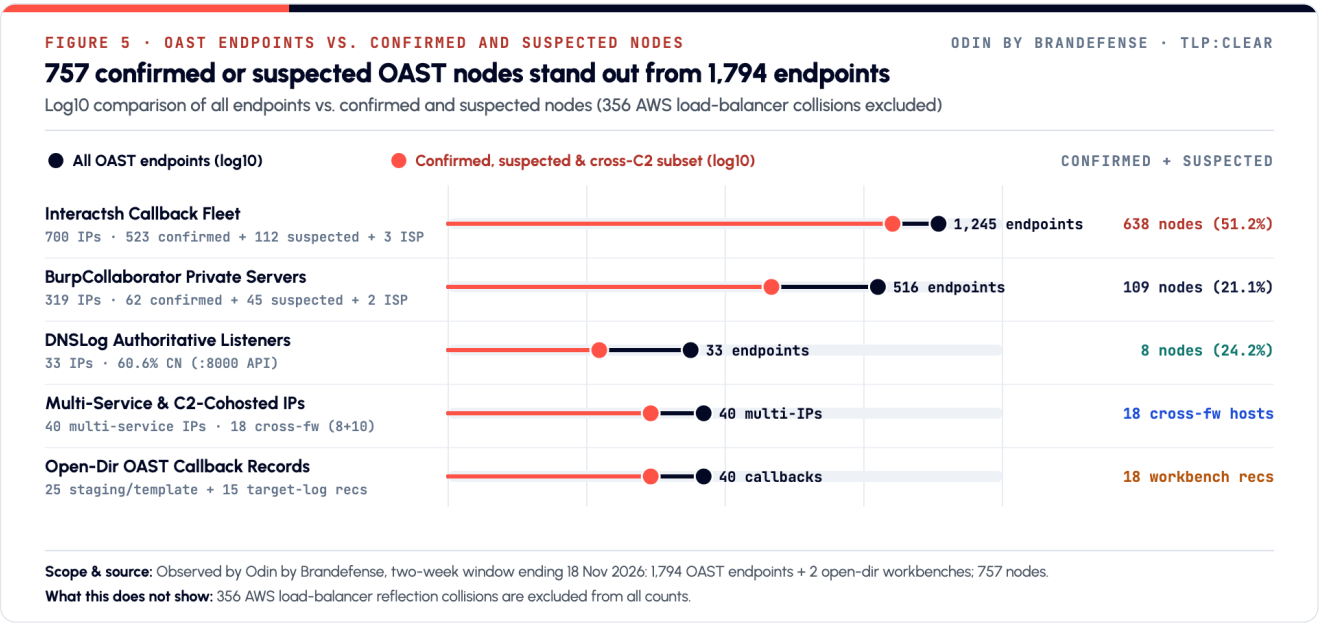


Figure 5: 757 confirmed or suspected OAST nodes stand out from 1,794 endpoints

What earns a block

A callback server goes on the block list only with independent offensive evidence. Two kinds qualify. The first is a C2 or reconnaissance framework on the same IP. The second is an exposed directory holding offensive scripts or exploit templates.

On that basis, 9 selected indicators are block-ready. That is 7 co-hosted callback or C2 nodes and 2 offensive workbenches. The 7 are six callback hosts from Chapter 2 plus 8.148.31[.]135 from Chapter 5. One more co-hosted host is withheld, because its Sliver port is only suspected and a red team cannot be ruled out.

What stays off the block list

Group	Size	Why not block	Action
AWS load-balancer reflections	356 records	They echo headers and only look like OAST servers	Exclude before any analysis
Standalone listeners and single-service `/16` hosts	7 standalone OAST and 7 `/16` entries in the selected set	No independent offensive evidence; often authorized testers	Watchlist; enrich with passive DNS
Consultancy Collaborator server	1	Carries an identifiable testing-firm certificate and no C2	Analyst review; check test scope
Listeners on dynamic subscriber lines	5 endpoints (3 Interactsh in Vietnam and China, 2 Burp Collaborator in the UK)	Home and small-business addresses change hands	Short watchlist that expires after 24 hours
Scanned third-party web application	1 host, 15 records	It logged incoming probes; it is a likely target	Withhold and exclude from threat lists
Your own approved scanners	Not counted	Internal DAST and red-team runners use the same tools	Allowlist by exact host, for scheduled windows

The same caution applies to the subnet cluster. Shared cloud ranges also host independent security researchers and commercial testing tenants.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

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

07 What Defenders Should Do

Within 24 hours

- [ ] Block the 9 block-ready endpoints in Chapter 9 at the perimeter, proxy and server egress.
- [ ] Hunt past DNS and HTTP logs for contact with those hosts and the eight workbench callback names in Chapter 4.
- [ ] Put the 18 watchlist and review entries in enrichment tables, not in firewall blocks.

Within 72 hours

- [ ] Block direct outbound UDP and TCP 53, HTTP and HTTPS (80, 443, 8000), SMTP (465, 587) and LDAP (389, 1389) from web, middleware and database subnets, except to explicit allowlists.
- [ ] Force all server DNS through internal Protective DNS forwarders that sinkhole `oast.*`, `interact[.]sh`, `oastify[.]com`, `burpcollaborator[.]net` and `dnslog[.]cn`.
- [ ] Register every internal DAST or red-team scanner IP and private Collaborator or Interactsh domain in a versioned allowlist before each test window.

Within 7 to 30 days

- [ ] Disable or restrict database network functions: `xp_dirtree` and `xp_subdirs` on SQL Server, `UTL_HTTP` and `UTL_INADDR` on Oracle, and `COPY ... FROM PROGRAM` on PostgreSQL.
- [ ] Deploy Suricata SIDs 9517001 to 9517010, the Sentinel server-egress query and the three Sigma rules.
- [ ] Alert on server lookups of 26 to 36 character random-looking labels.
- [ ] Recheck TLS certificate subjects and JARM values on elastic cloud IPs at regular intervals, since addresses get reassigned.

08

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 Sentinel query works 26

How the Sigma and YARA rules work 26

How to sort a host 27

08 Detection Rules

The full rules are in Appendix B. They use explicit host allowlists rather than wildcard exclusions.

Rule file	What it catches
`odin_oast_interactsh_burpcollab_dnslog.rules`	Suricata 7: ten signatures for callback DNS lookups, OAST server headers, DNSLog and Interactsh polling, and anonymous LDAP callbacks
`sentinel_internal_server_oast_egress_hunt.kql`	Microsoft Sentinel and Defender XDR: application and database server processes that reach OAST zones or the corroborated hosts, with follow-on activity
`sigma_server_tier_oast_dns_and_http_callback.yml`	Sigma (3 rules): server OAST lookups, database or web workers starting lookup tools, and unapproved OAST or offense bridge daemons
`odin_oast_payload_and_provisioning_scripts.yar`	YARA (4 rules): Interactsh and Collaborator server configs, DNSLog polling files, and multi-stack cloud provisioning files

How the Suricata signatures work

SID	Looks for	Where to deploy
9517001	DNS query to `.oast[.]pro`, `.live`, `.site`, `.online`, `.fun` or `.me`	DNS forwarder egress
9517002	DNS query for a long label under `.oastify[.]com`	DNS forwarder egress
9517003	HTTP response header `X-Interactsh-Version` or `Server: Interactsh-Server`	TLS inspection point
9517004	HTTP response header `BurpCollaboratorServer`	TLS inspection point
9517005	Outbound anonymous LDAPv3 bind to 389, 1389 or 10389	Perimeter TCP egress
9517006	HTTP request to `/getrecords` or `/getrecords.php` (BugScanTeam `/api/` routes not covered)	TLS inspection point
9517007	Interactsh client polling with `/poll?id=` and `&secret=`	TLS inspection point
9517008	DNS query for a 20 to 36 character label under `.interact[.]sh`	DNS forwarder egress
9517009	DNS query for a long label under `.burpcollaborator[.]net`	DNS forwarder egress
9517010	DNS query under `.dnslog[.]cn`	DNS forwarder egress

Bypass SIDs 9517001 to 9517004 and 9517007 to 9517010 only for documented scanner source IPs during scheduled test windows. Treat any SID 9517005 alert as high priority. Treat the DNS and header SIDs as high priority when they fire from a server subnet.

How the Sentinel query works

The query looks back 14 days. It watches network events from server processes such as `java`, `w3wp.exe`, `sqlservr.exe`, `postgres`, `mysqld`, `node` and `dotnet`. It skips listed scanner devices.

Signal	Fires when
Callback followed by follow-on	A server process reaches an OAST zone or a correlation-style name, then within 10 minutes starts a shell or `curl` / `wget`, or connects to a corroborated C2 or workbench endpoint
Standalone server callback	A server process reaches an OAST zone or correlation-style name
Egress to a corroborated host	A server process connects directly to one of the 11 corroborated hosts in the query's list
Port-only review	A server process reaches a public IP on 465, 587, 389, 1389 or 8000 with no OAST indicator and no approved relay

Fill in the approved scanner hostnames and the approved SMTP, LDAP or webhook relay IPs before use. The port-only signal is the weakest and is meant for review.

How the Sigma and YARA rules work

Rule	Fires when
Sigma 1: server OAST DNS query	A Windows web, middleware or database process queries an OAST zone, outside approved scanner hosts
Sigma 2: worker spawns lookup tool	A database or web worker starts `nslookup.exe`, `curl.exe`, `wget.exe` or `certutil.exe`, outside health checks
Sigma 3: OAST or offense daemon	A Linux host runs `interactsh-server`, `--collaborator-server`, `dnslog`, `cyberstrikeai` or `hexstrikeai`, outside sanctioned red-team hosts
`ODIN_Interactsh_SelfHosted_Server_Config`	An Interactsh server name appears with SMTP port settings or wildcard and listen settings
`ODIN_BurpCollaborator_Custom_Server_Config`	Collaborator JSON keys such as `serverDomain`, `polling` and `eventCapture` appear together
`ODIN_DNSLog_And_OAST_Polling_Descriptor`	A small poller file holds full `http` URLs for both `/getrecords` and `/getdomain`
`ODIN_IaC_MultiStack_Offensive_Burst_Manifest`	A Terraform file declares an Alibaba Cloud instance and lists quoted roles such as `cyberstrikeai`, `hexstrikeai`, `vshell` or `dnslog` with ports

The YARA rules were checked against four sanitized, non-executable descriptors and against BIND9 and Terraform baselines, with no false positives. The provisioning rule is built from an illustrative example; no such Terraform file was recovered.

How to sort a host

What the host shows	Action
A callback listener plus a C2 or recon framework on the same IP	Block
An exposed directory with offensive scripts or exploit templates	Block
A standalone listener, or a single-service host inside a cloud cluster	Watchlist and analyst review
A consultancy certificate with no C2	Analyst review
A listener on a dynamic subscriber line	24-hour watchlist
A third-party app logging incoming probes	Exclude from threat lists
A load-balancer reflection	Suppress
A documented internal scanner	Allow

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Block-ready endpoints 29

ATT&CK context 30

09 Indicators

Appendix A holds the 9 block-ready endpoints. The other 19 records in the selected set are withheld.

Type	Count
Block-ready endpoints (IP and port)	9
Watchlist and review entries (addresses withheld)	18
Load-balancer collision reference (withheld)	1
Distinct callback names in confirmed workbench files	8

Supporting fingerprints: the Interactsh JARM prefix `29d29d00029d29d00041d...`, the Burp Collaborator JARM prefix `28d28d28d00028d00042d...`, the VShell JARM prefix `29d29d16d...`, the CyberStrikeAI JARM prefix `40d40d40d...` and the certificate subject `CN=CyberStrikeAI`.

Block-ready endpoints

Endpoint	Role	Country	Hosting	Evidence	Earliest first seen on record
8.148.31[.]135:4446	Metasploit Meterpreter	CN	Alibaba Cloud	Meterpreter on 4446 and 443, Sliver on 8888	13 Nov 2026
49.232.168[.]207:8000	DNSLog	CN	Tencent	Confirmed DNSLog with VShell on 5030	13 Nov 2026
101.33.231[.]97:443	Interactsh	CN	Tencent	Cobalt Strike on 8080	13 Nov 2026
43.153.118[.]75:8300	Interactsh	US	Tencent	VShell on 8084 and 8094	13 Nov 2026
23.94.214[.]197:587	Interactsh	US	HostPapa	Interactsh on 443, 465, 587; ARTEX on 8787	13 Nov 2026
150.158.9[.]124:8000	DNSLog	CN	Tencent	ARL recon panel on 5003	13 Nov 2026
51.15.224[.]47:587	Interactsh	FR	Scaleway	Five-port Interactsh with ezXSS on 9443	13 Nov 2026
45.145.228[.]16:8080	Offensive open directory	HK	Cloudie	`/memshell/ExploitScriptEngineFactory.java` with an `oastify[.]com` callback	Before this window
47.101.51[.]235:8888	Offensive open directory	CN	Alibaba Cloud	`/EvilTemplate.java` and five `dnslog[.]cn` callbacks	Before this window

Records marked "Before this window" carry first-seen dates older than our observation. They are not the start of our observation, which covers the two weeks ending 18 November 2026.

The Sentinel query also lists two more corroborated hosts from earlier chapters: the workbench 144.172.99[.]199:8080 and the co-hosted workbench on 49.51.187[.]59:8090.

Withheld: the four `/16` cluster hosts, the eight `/16` siblings, and standalone Interactsh and Collaborator listeners with their certificate domains. Also withheld: the client log host, the five template mirrors, the consultancy Collaborator, the five subscriber addresses, the scanned Laravel target and the load-balancer collision host.

ATT&CK context

This activity maps to three MITRE ATT&CK tactics: Initial Access (blind probes and JNDI callbacks), Discovery (callback DNS lookups) and Command and Control (polling and co-hosted C2). See the Suricata table in Chapter 8 for which SID covers each.

10

CHAPTER 10

Scope and Limits

IN SHORT

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

10 Scope and Limits

- **Window.** The findings cover a two-week observation window ending 18 November 2026. Some records carry earlier first-seen dates; we do not use those as a tracking start.
- **Dual-use by default.** Interactsh, Burp Collaborator and DNSLog are standard testing tools. A listener alone says nothing about intent.
- **Candidates are leads.** The 1,039 candidates are hunting leads. Do not block them.
- **Proximity is not control.** Four services in one range show where they sat, not who runs them. No cross-host link was found.
- **Files are not outcomes.** Workbench scripts and templates show tooling and intent. We did not see any of them used against a target system.
- **No victim view.** This study looks at listeners from the outside. It has no logs from any target network.
- **Collisions removed.** 356 AWS load-balancer reflections are excluded from every count.
- **Rule testing.** Every published rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma and Microsoft's Kusto.Language analyzer, checked on 20 November 2026). The rules were not replayed against live traffic, so test and tune them in your own environment before production use.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. MITRE ATT&CK Enterprise tactics. https://attack.mitre.org/tactics/enterprise/ (verified 19 November 2026)
2. ProjectDiscovery. Interactsh repository. https://github.com/projectdiscovery/interactsh (verified 19 November 2026)
3. PortSwigger. Deploying a private Burp Collaborator server. https://portswigger.net/burp/documentation/collaborator/deploying (verified 19 November 2026)
4. BugScanTeam. DNSLog repository. https://github.com/BugScanTeam/DNSLog (verified 19 November 2026)
5. NSA and CISA. Selecting a Protective DNS Service. https://www.nsa.gov/Press-Room/News-Highlights/Article/Article/2523771/nsa-and-cisa-release-cybersecurity-information-on-protective-dns/ (verified 19 November 2026)
6. OWASP. Server-Side Request Forgery Prevention Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html (verified 19 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

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

INSIDE

A. Indicators of compromise (9) 36

B. Detection rules (4 files) 37

A Indicators of Compromise

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

ID	Type	Indicator	Port	Context	Action
R17-IOC-01	IPv4	8.148.31[.]135	4446	Metasploit / Meterpreter handler	BLOCK
R17-IOC-02	IPv4	49.232.168[.]207	8000	dnslog	BLOCK
R17-IOC-03	IPv4	101.33.231[.]97	443	interactsh	BLOCK
R17-IOC-04	IPv4	43.153.118[.]75	8300	interactsh	BLOCK
R17-IOC-05	IPv4	23.94.214[.]197	587	interactsh	BLOCK
R17-IOC-06	IPv4	150.158.9[.]124	8000	dnslog	BLOCK
R17-IOC-07	IPv4	51.15.224[.]47	587	interactsh	BLOCK
R17-IOC-08	IPv4	45.145.228[.]16	8080	opendir_oast	BLOCK
R17-IOC-09	IPv4	47.101.51[.]235	8888	opendir_oast	BLOCK

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_oast_interactsh_burpcollab_dnslog.rules

SURICATA

```
# Odin by Brandefense - Report #17 Suricata 7.x OAST Callback Fabrics & Polling Suite
# Classification: TLP:CLEAR
# Scope Note: Network DNS, HTTP/TLS offload, and anonymous LDAP callback signatures (sid:9517001-9517010).
#             Pair OAST callback alerts with explicit sanctioned VAPT scanner source-IP allowlists.

# 1. Outbound DNS Query to Public ProjectDiscovery .oast.* Callback Zones
alert dns $HOME_NET any -> any 53 (msg:"ODIN-OAST Outbound DNS Query to Public Interactsh .oast.* Zone";
dns.query; content:".oast."; nocase; fast_pattern; pcre:"/\.oast\.(pro|live|site|online|fun|me)$/i";
metadata:created_at 2026_11_18, tactic discovery; classtype:bad-unknown; sid:9517001; rev:2;)

# 2. Outbound DNS Query to PortSwigger oastify.com Callback Zone
alert dns $HOME_NET any -> any 53 (msg:"ODIN-OAST Outbound DNS Query to PortSwigger oastify.com Zone"; dns.query;
content:".oastify.com"; nocase; fast_pattern; pcre:"/[a-z0-9]{16,40}\.oastify\.com$/i"; metadata:created_at
2026_11_18, tactic discovery; classtype:bad-unknown; sid:9517002; rev:2;)

# 3. Inbound HTTP Response from Self-Hosted Interactsh Server (X-Interactsh-Version / Interactsh-Server)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-OAST Self-Hosted Interactsh Server Response Header";
flow:established,from_server; http.header; content:"Interactsh"; nocase; fast_pattern; pcre:"/^(X-Interactsh-
Version|Server:\s*Interactsh-Server)/mi"; metadata:created_at 2026_11_18, tactic command_and_control;
classtype:trojan-activity; sid:9517003; rev:1;)

# 4. Inbound HTTP Response from Private BurpCollaborator Server (Server: BurpCollaboratorServer)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-OAST Private BurpCollaborator Server Response Header";
flow:established,from_server; http.header; content:"BurpCollaboratorServer"; fast_pattern; metadata:created_at
2026_11_18, tactic command_and_control; classtype:trojan-activity; sid:9517004; rev:1;)

# 5. Outbound Anonymous LDAPv3 Bind Callback Over Non-Standard External JNDI Ports
alert tcp $HOME_NET any -> $EXTERNAL_NET [389,1389,10389] (msg:"ODIN-OAST Outbound Anonymous LDAPv3 Callback
Verification"; flow:established,to_server; content:"|30 0c 02 01 01 60 07 02 01 03 04 00 80 00|"; depth:14;
fast_pattern; metadata:created_at 2026_11_18, tactic initial_access; classtype:attempted-admin; sid:9517005; rev:
1;)

# 6. dnslog.cn-compatible DNSLog HTTP API Record Polling (/getrecords, /getrecords.php; BugScanTeam /api/<type>/
<domain>/<hash>/ routes are not covered)
alert http any any -> $EXTERNAL_NET any (msg:"ODIN-OAST DNSLog HTTP API Callback Record Polling (/getrecords)";
flow:established,to_server; http.uri; content:"getrecords"; nocase; fast_pattern; pcre:"/^\/getrecords(?:\.php)?
(?:[/?]|$)/i"; metadata:created_at 2026_11_18, tactic command_and_control; classtype:trojan-activity; sid:
9517006; rev:2;)

# 7. Interactsh Client Encrypted Correlation Polling Request (/poll?id=)
alert http any any -> $EXTERNAL_NET any (msg:"ODIN-OAST Interactsh Client Correlation Polling Request (/poll?
id=)"; flow:established,to_server; http.uri; content:"/poll?id="; nocase; fast_pattern; content:"&secret=";
nocase; distance:0; metadata:created_at 2026_11_18, tactic command_and_control; classtype:trojan-activity; sid:
9517007; rev:1;)

# 8. Outbound DNS Query or High-Entropy Subdomain Exfiltration Over interact.sh
alert dns $HOME_NET any -> any 53 (msg:"ODIN-OAST Outbound DNS Query or Exfiltration Over interact.sh";
dns.query; content:".interact.sh"; nocase; fast_pattern; pcre:"/^(?:[a-z0-9_-]{1,63}\.)?[a-z0-9]
{20,36}\.interact\.sh$/i"; metadata:created_at 2026_11_18, tactic command_and_control; classtype:trojan-activity;
sid:9517008; rev:2;)

# 9. Outbound DNS Query to PortSwigger burpcollaborator.net Callback Zone
alert dns $HOME_NET any -> any 53 (msg:"ODIN-OAST Outbound DNS Query to PortSwigger burpcollaborator.net Zone";
dns.query; content:".burpcollaborator.net"; nocase; fast_pattern; pcre:"/[a-z0-9]{16,40}\.burpcollaborator\.net$/
i"; metadata:created_at 2026_11_18, tactic discovery; classtype:bad-unknown; sid:9517009; rev:1;)
```

```
# 10. Outbound DNS Query to Public dnslog.cn Callback Zone
alert dns $HOME_NET any -> any 53 (msg:"ODIN-OAST Outbound DNS Query to Public dnslog.cn Zone"; dns.query;
content:".dnslog.cn"; nocase; fast_pattern; pcre:"/[a-z0-9_-]{4,40}\.dnslog\.cn$/i"; metadata:created_at
2026_11_18, tactic discovery; classtype:bad-unknown; sid:9517010; rev:1;)
```

odin_oast_payload_and_provisioning_scripts.yar

YARA

```
/*
   Odin by Brandefense - Report #17 YARA OAST Server & Cloud Burst Provisioning Suite
   Classification: TLP:CLEAR
   Scope: Detects self-hosted Interactsh server descriptors, private BurpCollaborator JSON configs,
          DNSLog / OAST callback polling configurations, and multi-stack IaC cloud burst manifests.
*/

rule ODIN_Interactsh_SelfHosted_Server_Config
{
    meta:
        author = "Odin by Brandefense"
        report = "Monograph 17 - OAST Callback Fabrics and Cloud Bursts"
        classification = "TLP:CLEAR"
        tactic = "Command and Control"
    strings:
        $i1 = "Interactsh-Server" ascii wide nocase
        $i2 = "interactsh-server" ascii wide nocase
        $p1 = "smtp-port:" ascii wide nocase
        $p2 = "smtps-port:" ascii wide nocase
        $w1 = "wildcard: true" ascii wide nocase
        $w2 = "listen-ip:" ascii wide nocase
    condition:
        filesize < 512KB and (($i1 or $i2) and ($p1 or $p2) and ($w1 or $w2))
}

rule ODIN_BurpCollaborator_Custom_Server_Config
{
    meta:
        author = "Odin by Brandefense"
        report = "Monograph 17 - OAST Callback Fabrics and Cloud Bursts"
        classification = "TLP:CLEAR"
        tactic = "Command and Control"
    strings:
        $b1 = "\"serverDomain\"" ascii wide
        $b2 = "\"polling\"" ascii wide
        $c1 = "\"eventCapture\"" ascii wide
        $c2 = "\"workerThreads\"" ascii wide
        $d1 = "\"publicAddress\"" ascii wide
        $d2 = "\"localAddress\"" ascii wide
        $e1 = "\"dns\"" ascii wide
    condition:
        filesize < 512KB and ($b1 and $b2 and ($c1 or $c2) and ($d1 or $d2) and $e1)
}

rule ODIN_DNSLog_And_OAST_Polling_Descriptor
{
    meta:
        author = "Odin by Brandefense"
        report = "Monograph 17 - OAST Callback Fabrics and Cloud Bursts"
        classification = "TLP:CLEAR"
        tactic = "Discovery"
    strings:
        $d1 = /https?:\/\/[^ \t\r\n"']{1,80}\/getrecords/ ascii nocase
        $d2 = /https?:\/\/[^ \t\r\n"']{1,80}\/getdomain/ ascii nocase
        $z1 = ".dnslog.cn" ascii wide nocase
        $z2 = ".oast.pro" ascii wide nocase
        $z3 = ".oast.site" ascii wide nocase
        $z4 = ".oast.online" ascii wide nocase
    condition:
        filesize < 16KB and ($d1 and $d2 and 1 of ($z*))
```

```
}

rule ODIN_IaC_MultiStack_Offensive_Burst_Manifest
{
    meta:
        author = "Odin by Brandefense"
        report = "Monograph 17 - OAST Callback Fabrics and Cloud Bursts"
        classification = "TLP:CLEAR"
        tactic = "Command and Control"
    strings:
        $iac1 = /resource[ \t]+"alicloud_instance"/ ascii
        $s1 = /"cyberstrikeai:[0-9]{2,5}"/ ascii nocase
        $s2 = /"hexstrikeai:[0-9]{2,5}"/ ascii nocase
        $s3 = /"vshell:[0-9]{2,5}"/ ascii nocase
        $s4 = /"dnslog:[0-9]{2,5}"/ ascii nocase
    condition:
        filesize < 64KB and $iac1 and 3 of ($s*)
}
```

sentinel_internal_server_oast_egress_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Report #17 Sentinel / Defender XDR Internal Server OAST Egress & C2 Handoff Hunt
// Classification: TLP:CLEAR
// Scope: Hunts application and database server egress to OAST callback infrastructure and joins with WebSocket
C2 handoffs.
let Lookback = 14d;
let HandoffWindow = 10m;
let ApprovedVaptDevices = dynamic(["vapt-scanner-01.corp.example", "dast-runner-01.corp.example"]);
let ApprovedOutboundRelayIPs = dynamic(["198.51.100.25", "203.0.113.50"]);
let KnownOastSuffixes = dynamic([
    ".oast.pro", ".oast.site", ".oast.online", ".oast.live", ".oast.fun", ".oast.me", ".interact.sh",
".oastify.com", ".burpcollaborator.net", ".dnslog.cn"
]);
let VerifiedR17HostileNodes = dynamic([
    "8.148.31.135", "49.232.168.207", "101.33.231.97", "43.153.118.75", "23.94.214.197",
    "150.158.9.124", "51.15.224.47", "49.51.187.59", "144.172.99.199", "45.145.228.16", "47.101.51.235"
]);
let VerifiedR17C2Endpoints = dynamic([
    "8.148.31.135:4446", "8.148.31.135:443", "8.148.31.135:8888",
    "49.232.168.207:5030", "101.33.231.97:8080", "43.153.118.75:8084", "43.153.118.75:8094",
"23.94.214.197:8787",
    "150.158.9.124:5003", "51.15.224.47:9443", "49.51.187.59:8090", "144.172.99.199:8080", "45.145.228.16:8080",
"47.101.51.235:8888"
]);
let ServerProcesses = dynamic([
    "java", "java.exe", "w3wp.exe", "sqlservr.exe", "postgres", "mysqld", "oracle.exe", "node", "node.exe", "php-
fpm", "python3", "dotnet", "dotnet.exe"
]);
let ServerEgressAnnotated =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedVaptDevices))
    | where InitiatingProcessFileName in~ (ServerProcesses)
    | extend CleanHost = trim_end(@"\.", tolower(extract(@"^(?:https?://)?([^/:?#]+)", 1, coalesce(RemoteUrl,
"")))))
    | extend IsKnownOastZone = (
        CleanHost endswith ".oast.pro" or CleanHost endswith ".oast.site" or CleanHost endswith ".oast.online"
        or CleanHost endswith ".oast.live" or CleanHost endswith ".oast.fun" or CleanHost endswith ".oast.me"
        or CleanHost endswith ".interact.sh" or CleanHost endswith ".oastify.com" or CleanHost endswith
".burpcollaborator.net" or CleanHost endswith ".dnslog.cn"
    )
    | extend IsCorrelationSubdomain = CleanHost matches regex @"^(?:[a-z0-9_-]{1,63}\.)?[a-z0-9]{28,36}\.[a-
z0-9.-]+\.[a-z]{2,10}$"
    | extend IsOastDomain = (IsKnownOastZone or IsCorrelationSubdomain)
    | extend IsHostileNodeEgress = RemoteIP in (VerifiedR17HostileNodes)
    | extend HasOastIndicator = (IsOastDomain or IsHostileNodeEgress)
    | extend IsPortOnlyCandidate = (not(HasOastIndicator) and RemotePort in (465, 587, 389, 1389, 8000) and
RemoteIPType == "Public" and not(RemoteIP in (ApprovedOutboundRelayIPs)))
```

```
    | where HasOastIndicator or IsPortOnlyCandidate
    | project DeviceName, OastTime = Timestamp, ServerProc = InitiatingProcessFileName, HasOastIndicator,
IsOastDomain, IsHostileNodeEgress, IsPortOnlyCandidate,
              OastTarget = strcat(RemoteIP, ":", tostring(RemotePort), " (", iff(isempty(CleanHost), "direct_ip",
CleanHost), ")");
let ServerOastCallbacks = ServerEgressAnnotated | where IsOastDomain;
let ServerHostileNodeEgress = ServerEgressAnnotated | where IsHostileNodeEgress and not(IsOastDomain);
let ServerPortOnlyCandidates = ServerEgressAnnotated | where IsPortOnlyCandidate;
let FollowOnC2OrShells =
    union isfuzzy=true
    (
        DeviceProcessEvents
        | where Timestamp >= ago(Lookback)
        | where InitiatingProcessFileName in~ (ServerProcesses)
        | where FileName in~ ("sh", "bash", "cmd.exe", "powershell.exe", "pwsh", "curl", "curl.exe", "wget",
"wget.exe")
        | project DeviceName, FollowTime = Timestamp, FollowType = "Server_Child_Shell",
                  FollowDetail = strcat(InitiatingProcessFileName, " -> ", FileName, ": ", ProcessCommandLine)
    ),
    (
        DeviceNetworkEvents
        | where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess"
        | extend Endpoint = strcat(RemoteIP, ":", tostring(RemotePort))
        | where Endpoint in (VerifiedR17C2Endpoints)
        | project DeviceName, FollowTime = Timestamp, FollowType = "Corroborated_R17_C2_Or_Workbench",
                  FollowDetail = strcat(InitiatingProcessFileName, " -> ", Endpoint)
    );
union isfuzzy=true
(
    ServerOastCallbacks
    | join kind=inner (FollowOnC2OrShells) on DeviceName
    | where FollowTime > OastTime and FollowTime <= (OastTime + HandoffWindow)
    | project Timestamp = FollowTime, DeviceName, HuntSignal = strcat("OAST_Callback_To_", FollowType),
              InitiatingProcess = ServerProc, ArtifactDetail = strcat("OAST=", OastTarget, " => ", FollowDetail)
),
(
    ServerOastCallbacks
    | project Timestamp = OastTime, DeviceName, HuntSignal = "Standalone_Server_Tier_OAST_Egress",
              InitiatingProcess = ServerProc, ArtifactDetail = OastTarget
),
(
    ServerHostileNodeEgress
    | project Timestamp = OastTime, DeviceName, HuntSignal = "Server_Egress_To_Corroborated_R17_Node",
              InitiatingProcess = ServerProc, ArtifactDetail = OastTarget
),
(
    ServerPortOnlyCandidates
    | project Timestamp = OastTime, DeviceName, HuntSignal = "Candidate_Server_Port_Only_Egress_Review",
              InitiatingProcess = ServerProc, ArtifactDetail = OastTarget
)
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), SampleIndicators =
make_set(ArtifactDetail, 5)
    by DeviceName, HuntSignal, InitiatingProcess
| order by EventCount desc
```

sigma_server_tier_oast_dns_and_http_callback.yml

SIGMA

```
title: Application or Database Server Out-of-Band OAST DNS Query
id: 7a4c9e17-31b2-4d8f-a617-170000000001
status: experimental
description: Detects outbound DNS queries from internal web, middleware, or database tiers toward public or
verified self-hosted OAST callback zones.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.initial_access
  - attack.discovery
logsource:
```

```
    category: dns_query
    product: windows
  detection:
    selection_proc:
      Image|endswith:
        - '\java.exe'
        - '\w3wp.exe'
        - '\sqlservr.exe'
        - '\postgres.exe'
        - '\mysqld.exe'
        - '\node.exe'
        - '\php-cgi.exe'
        - '\dotnet.exe'
    selection_oast:
      QueryName|endswith:
        - '.oast.pro'
        - '.oast.site'
        - '.oast.online'
        - '.oast.live'
        - '.oast.fun'
        - '.oast.me'
        - '.interact.sh'
        - '.oastify.com'
        - '.burpcollaborator.net'
        - '.dnslog.cn'
    filter_vapt_host:
      ComputerName:
        - 'VAPT-SCANNER-01'
        - 'DAST-RUNNER-01'
    condition: selection_proc and selection_oast and not filter_vapt_host
  falsepositives:
    - Authorized dynamic application security testing runs targeting staging servers without host exclusions
  level: high
  ---
  title: RDBMS or Web Worker Spawning Out-of-Band Lookup Utility
  id: 7a4c9e17-31b2-4d8f-a617-170000000002
  status: experimental
  description: Detects database or web server worker processes invoking nslookup, curl, wget, or certutil to
  validate blind command or SSRF reachability.
  author: Odin by Brandefense
  date: 2026-11-18
  tags:
    - attack.discovery
    - attack.command_and_control
  logsource:
    category: process_creation
    product: windows
  detection:
    selection_parent:
      ParentImage|endswith:
        - '\sqlservr.exe'
        - '\w3wp.exe'
        - '\java.exe'
        - '\tomcat9.exe'
        - '\postgres.exe'
    selection_child:
      Image|endswith:
        - '\nslookup.exe'
        - '\curl.exe'
        - '\wget.exe'
        - '\certutil.exe'
        - '\ping.exe'
    # No command-line substring exception: suppress approved health checks by exact host in the SIEM.
    condition: selection_parent and selection_child
  falsepositives:
    - Legacy application monitoring scripts running directly under web worker pools
  level: high
  ---
  title: Self-Hosted OAST Collector or Autonomous Offense Bridge Execution
  id: 7a4c9e17-31b2-4d8f-a617-170000000003
  status: experimental
  description: Detects unauthorized execution of self-hosted Interactsh, private BurpCollaborator, DNSLog,
```

```
  CyberStrikeAI, or HexStrikeAI server daemons on cloud Linux instances.
  author: Odin by Brandefense
  date: 2026-11-18
  tags:
    - attack.discovery
    - attack.command_and_control
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_bin:
      Image|endswith:
        - '/interactsh-server'
        - '/cyberstrikeai'
        - '/hexstrikeai'
    selection_cli:
      CommandLine|contains:
        - '--collaborator-server'
        - 'collaborator-config='
        - 'dnslog/manage.py runserver'
        - 'interactsh-server -domain'
    # No command-line substring exception: suppress sanctioned OAST servers by exact host in the SIEM.
    condition: selection_bin or selection_cli
  falsepositives:
    - Sanctioned internal offensive security infrastructure registered in asset inventory
  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/out-of-band-callback-fabrics/
Short Read · brandefense.io/odin-research/blog/callback-servers-are-everywhere-a-few-sit-next-to-attack/
Defense Kit · brandefense.io/odin-research/defense-kit/