ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 13 / 30 · DETECTION CALIBRATION

C2 Threat-Feed Collisions

2,081 False Positives in Enterprise Stacks

How single-signal C2 rules mislabel mTLS gateways, cloud load balancers, PHP servers, MySQL databases and test fixtures

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

25,482
Deduplicated indicators reviewed: 24,660 C2 endpoints plus 822 open directories

2,081
C2 rule collisions on benign production servers (8.2%), across 1,958 unique IPv4 hosts

1,592
MySQL servers on port 3306 labelled confirmed Daybreak C2

356
AWS Application Load Balancer endpoints labelled Interactsh OAST servers

What Is in This Report. 10 chapters and two appendices.

How single-signal C2 rules mislabel mTLS gateways, cloud load balancers, PHP servers, MySQL databases and test fixtures

01	Executive Summary	04
02	Why Single-Signal Rules Collide	07
03	Automaker mTLS Gateways and Cloud Load Balancers	10
04	PHP 8.5, MySQL 3306 and Test Fixtures	14
05	How Much of a Raw Feed Is Safe to Block	19
06	What Defenders Should Do	23
07	Detection Rules	25
08	Indicators	28
09	Scope and Limits	31
10	Sources	33
A	Indicators of compromise (8)	35
B	Detection rules (4 files)	36

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense found 2,081 cases where a C2 detection rule labelled a benign production server as command-and-control. The hits covered automaker mTLS gateways, AWS load balancers, PHP 8.5 web servers and MySQL databases. Another 135 secret-scanner alerts came from open-source test files. Each rule matched one surface trait. Do not let a threat feed auto-block any address until certificate, protocol and network checks agree.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense found 2,081 cases where a C2 detection rule labelled a benign production server as command-and-control. The hits covered automaker mTLS gateways, AWS load balancers, PHP 8.5 web servers and MySQL databases. Another 135 secret-scanner alerts came from open-source test files. Each rule matched one surface trait. Do not let a threat feed auto-block any address until certificate, protocol and network checks agree.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 04

01 Executive Summary

Many threat feeds label a server as command-and-control (C2) because it matches one trait: a TLS fingerprint, an HTTP status code, a response header or an open port. When a SOC wires such a feed straight into firewall or EDR block lists, every weak rule becomes a way to cut off its own partners. This report shows where that happens, why, and which checks tell a real C2 server from a collision.

1. **Single-trait rules produced 2,081 false C2 hits across four production stacks.**
   Six automaker mTLS gateways were labelled Emotet, one at a raw score of 97. 356 AWS load balancers were labelled Interactsh and 127 PHP 8.5 web servers ezXSS. Another 1,592 MySQL servers on port 3306 were labelled confirmed Daybreak C2.

2. **A shared AWS TLS fingerprint makes load balancers look like OAST servers.**
   189 of the 356 load-balancer hits share the exact JARM value seen on AWS-hosted Interactsh servers. Only a certificate issuer check plus an application-layer check separates the two. An Interactsh server is a dual-use testing tool, so even a real one is a hunting lead, not proof of hostile use.

3. **Test fixtures and home broadband create a second wave of bad alerts.**
   Of 5,576 secret findings in exposed directories, 135 came from unit tests, skill docs and public prompt archives, including 41 rated critical. Separately, 197 C2 hits sat on residential and mobile networks, where addresses change hands. Those need short watchlists, not permanent blocks.

Three Decisions for Leaders

1. **Stop single-signal auto-blocking.** No feed indicator should reach a firewall or EDR block list unless protocol behaviour, certificate issuer and network type all point the same way.
2. **Give each kind of indicator its own lifetime.** Block corroborated C2 on stable hosting for 14 days. Hunt dual-use OAST servers and suspected C2 for 7 days. Watch subscriber ISP addresses for 24 to 72 hours. Keep single-signal candidates for correlation only.
3. **Read file paths before raising secret alerts.** Put live AI agent session folders first. Then filter unit tests, skill reference docs, public prompt archives and known vendor example keys, so analysts see real leaks.

The Study in Numbers

Number	What it is
25,482	Deduplicated indicators reviewed: 24,660 C2 endpoints plus 822 open directories
2,081	C2 rule collisions on benign production servers (8.2%), across 1,958 unique IPv4 hosts
1,592	MySQL servers on port 3306 labelled confirmed Daybreak C2
356	AWS Application Load Balancer endpoints labelled Interactsh OAST servers
135	Secret-scanner hits from open-source test fixtures, out of 5,576 secret findings
3,916	Confirmed, separately corroborated C2 indicators fit for a block list (15.4%)
197	C2 hits on dynamic residential or mobile networks, moved to 24 to 72 hour watchlists

How to use this report.

Security leaders can read Chapter 1, Chapter 5 and Chapter 6. Detection engineers and SOC analysts should start with Chapters 3, 4 and 7. CTI and feed owners will find the root causes in Chapter 2 and the indicator counts in Chapter 8.

02

CHAPTER 02

Why Single-Signal Rules Collide

IN SHORT

Every collision in this study has the same root cause: a rule that matches one surface trait. TLS fingerprints, status codes, headers and port banners describe the server software, not the intent of whoever runs it.

INSIDE THIS CHAPTER

One trait is not an identity 07
The five collision classes 08
Why this matters for automated blocking 08

02 Why Single-Signal Rules Collide

One trait is not an identity

A JARM value describes how a TLS library answers a handshake. An HTTP 400 page describes how a web server rejects a request. An `X-Powered-By` header names a runtime version. A port banner names a database. Thousands of legitimate servers share each of these traits with some C2 framework.

When a rule matches only one of them, every benign server with the same trait inherits the C2 label. That is the pattern behind all 2,081 C2 collisions and all 135 secret-scanner collisions in this report.

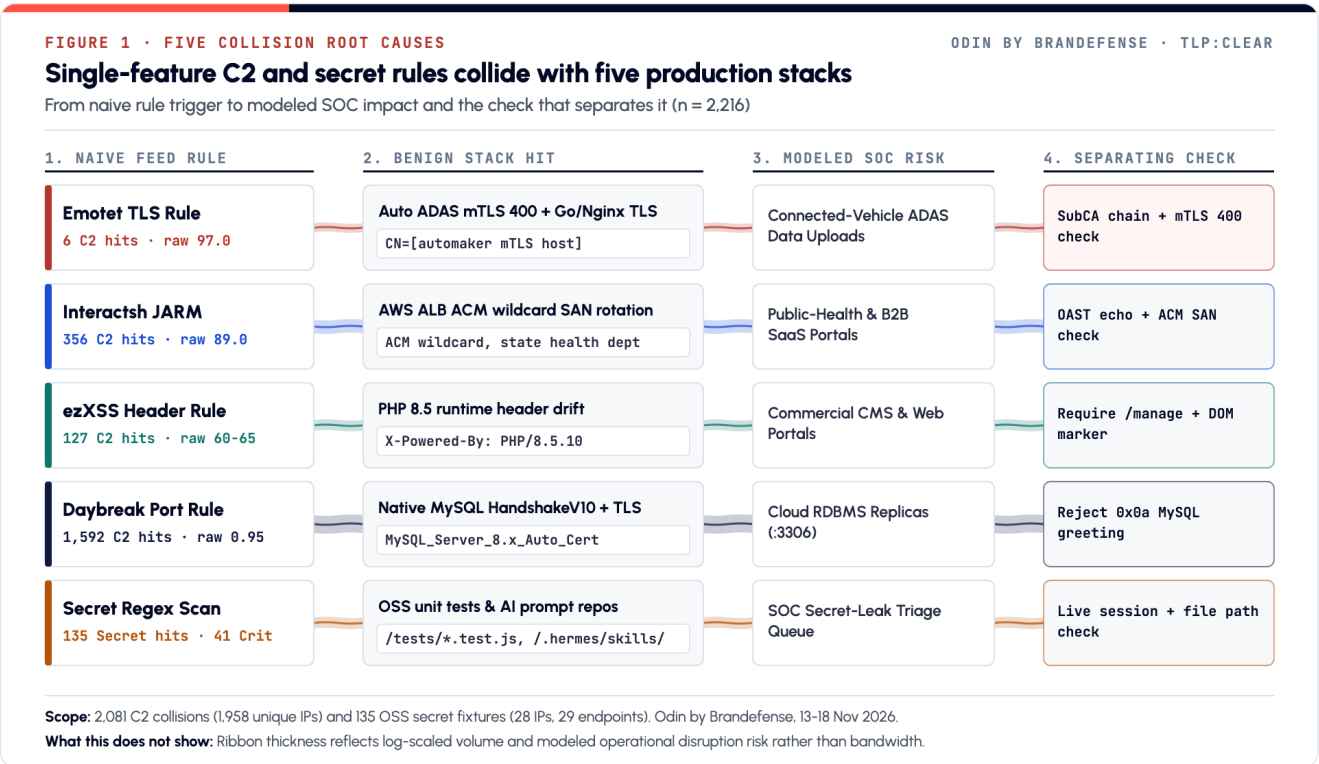


Figure 1: Single-feature C2 and secret rules collide with five production stacks

The five collision classes

The 2,081 C2 collisions sit on 1,958 unique IPv4 hosts. The 135 secret-scanner collisions sit on 28 unique IPs across 29 open-directory endpoints. Together they make 2,216 collisions.

Class	Rule that fired	Rows (unique IPs)	Raw label and score	Benign stack it hit	What separates it from C2
1	`emotet`	6 (5)	1 confirmed at 97.0, 5 suspected at 80.0	A Chinese automaker's ADAS and corporate mTLS gateways	Private enterprise sub-CA issuer plus the mTLS "no client certificate" HTTP 400 page
2	`interactsh`	356 (261)	356 suspected at 89.0	AWS Application Load Balancers with ACM wildcard certificates	A real Interactsh server reflects test IDs and listens on several ports
3	`ezxss`	127 (101)	106 suspected at 65.0, 21 at 60.0	Commercial CMS and portals upgraded to `PHP/8.5.10`	No `/manage/login` panel and no `ezxss.js` collector script
4	`daybreak`	1,592 (1,592)	1,592 confirmed at 0.95	Cloud MySQL 5.7, 8.0, 8.4 and 9.1 on `TCP/3306`	MySQL greeting byte `0x0a` and an auto-generated MySQL certificate
5	Secret regex	135 (28)	41 critical, 15 high, 79 medium	Unit tests, skill docs, public AI prompt archives	File path context, known example keys, no live session state

The raw labels and scores are those carried by the detection feed before any checks. One host triggered both the ezXSS and the Daybreak rule, so the class-level host counts add up to 1,959, not 1,958.

The framework names in the "Rule that fired" column are feed labels. They are not attribution of any benign host to any threat actor.

Why this matters for automated blocking

The most dangerous false positives are not low-scoring candidates. They are the high-scoring ones, here 89 to 97, because SOAR playbooks often auto-block anything above a fixed score.

A block on the automaker gateways could cut vehicle data uploads for a supplier or fleet operator. A block on the load balancers could cut a state public-health portal or a logistics SaaS. A block on the MySQL servers could break cloud database replicas. These are modeled risks: we did not observe any actual outage.

CHAPTER 03

Automaker mTLS Gateways and Cloud Load Balancers

IN SHORT

362 high-scoring collisions came from TLS-layer rules. Six automaker gateways looked like Emotet, and 356 AWS load balancers looked like Interactsh, because their TLS answers matched.

INSIDE THIS CHAPTER

Class 1: automaker mTLS gateways labelled Emotet 10
Class 2: AWS load balancers labelled Interactsh 11
Could a load balancer front a real Interactsh server? 12
Do not allowlist all of AWS 12

03 Automaker mTLS Gateways and Cloud Load Balancers

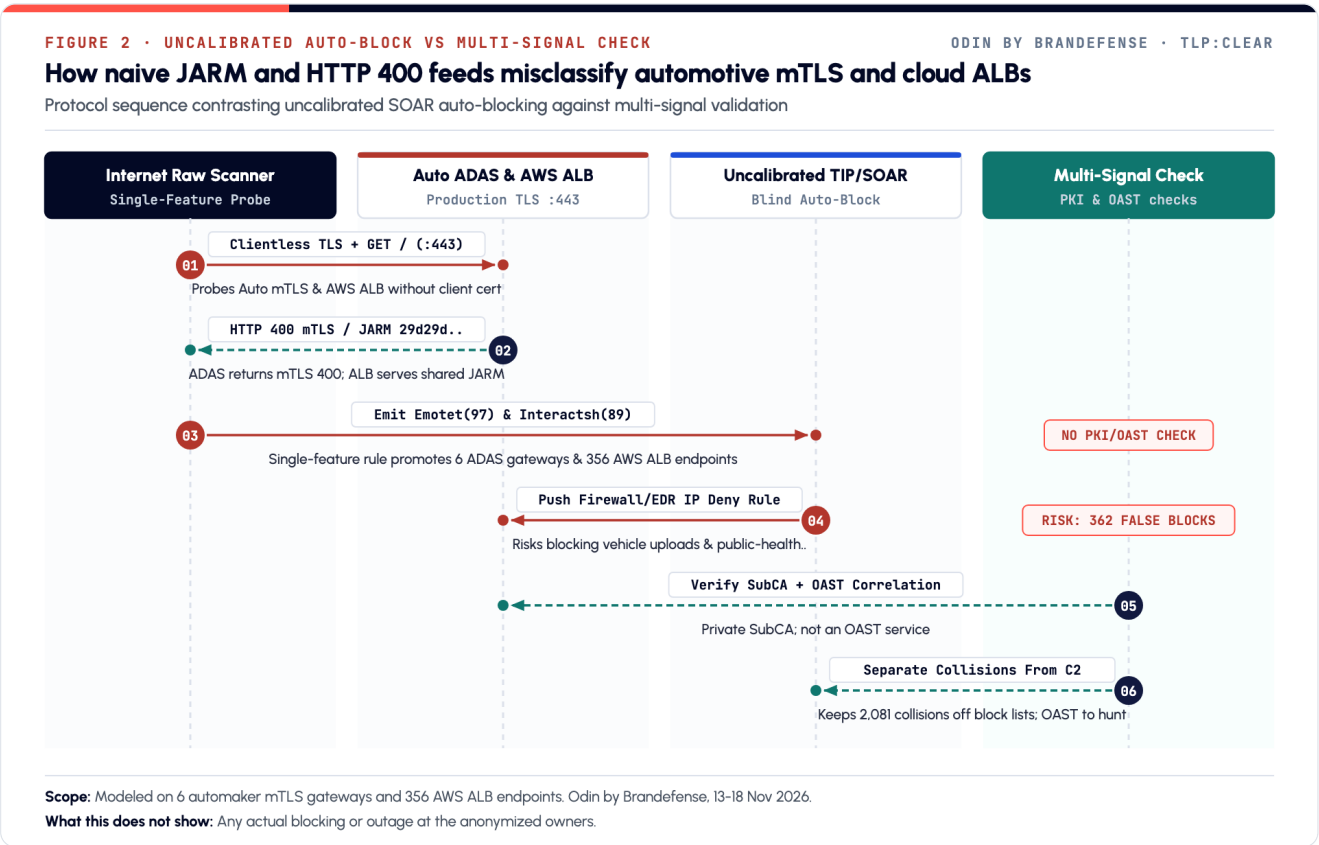


Figure 2: How naive JARM and HTTP 400 feeds misclassify automotive mTLS and cloud ALBs

Class 1: automaker mTLS gateways labelled Emotet

On 18 November 2026, an Emotet rule labelled one gateway of a Chinese automaker as confirmed Emotet C2 at a raw score of 97.0. It labelled five more of the automaker's gateways as suspected Emotet at 80.0. One primary and one companion gateway sit on Alibaba Cloud. The other four sit on Huawei Cloud, on ports 443, 8080 and 80.

The automaker's name, domains and addresses are withheld. The gateways are benign and must not enter any block list.

What the gateway shows	What it means
Leaf certificate for an ADAS mTLS host, issued by a private enterprise sub-CA	Issuer and subject differ: this is a corporate PKI, not a self-signed C2 certificate
Five companion gateways with a wildcard certificate for the automaker's telematics domain	Same corporate PKI across the fleet
`ssl_verify_client on` in Nginx in front of Go services	The gateway demands a client certificate (mutual TLS)
HTTP `400 Bad Request` with "No required SSL certificate was sent"	The normal answer to any scanner that has no client certificate

ADAS stands for Advanced Driver-Assistance Systems. These gateways carry wireless ADAS and connected-vehicle uploads.

Why the Emotet rule fired.

Emotet Epoch 4 and Epoch 5 used Nginx reverse proxies. They return short HTTP 400 or 404 answers over stock Linux OpenSSL when a POST lacks a valid payload. A rule that matches "Nginx 400 on a cloud network" fires on both.

The rule did not check for the "No required SSL certificate was sent" text. It did not check whether the issuer differs from the subject. Either check would have cleared the gateways.

How real Emotet C2 looked.

The three confirmed Emotet servers in our indicator set use self-signed certificates, where issuer equals subject. Examples are `CN=greenedgewealth-zesle`, `CN=serv-01.nexgeninter[.]net` and `CN=Callsign`. All three sit on DigitalOcean on port 443.

Class 2: AWS load balancers labelled Interactsh

On 17 November 2026, 356 endpoints raised suspected Interactsh alerts at a raw score of 89.0. They sat on 261 unique Amazon IPv4 addresses: 259 rows on port 443 and 97 on port 80.

Interactsh is ProjectDiscovery's out-of-band application security testing (OAST) server. It catches DNS, HTTP and SMTP callbacks from test payloads. Operators often run it on AWS EC2 with a wildcard certificate, so that random correlation subdomains resolve over HTTPS.

The collision behind the 356 rows came from eight anonymized certificate subjects.

Owner sector (anonymized)	Collision rows
Enterprise transportation management SaaS	116
Commercial analytics and AI SaaS	109
State public-health agency, non-production cloud	83
Cloud provider edge routing	22
Financial payment and accounting integration SaaS	8
Commercial media web platform	8
Regional logistics and storage portal	6
Field-service dispatch SaaS	4

Why the Interactsh rule fired.

189 of the 356 rows return JARM `29d29d00029d29d00041d41d00041dcb3f8752a2d70d1dbf446fc46ab5df96`. That is the exact 62-character JARM seen on active AWS-hosted Interactsh servers. The value reflects the AWS TLS stack, not the tenant.

AWS Application Load Balancers (`awselb/2.0`) scale across short-lived EC2 address pools. They serve Amazon Certificate Manager (ACM) wildcard certificates and a generic default answer to scans that hit a bare IP. A JARM plus wildcard-certificate rule turned eight enterprise domains into 356 false OAST alerts.

What a real Interactsh server shows instead.

It reflects the correlation ID of a test request back to the caller. It listens on several ports at once, typically 53, 80, 443, 465 and 587. It exposes a `/register` API.

Could a load balancer front a real Interactsh server?

Yes, in principle. ALB listener rules can route one host name or path to a backend container, including an Interactsh container behind an enterprise wildcard certificate.

A load balancer that answers a bare-IP request with its default `awselb/2.0` page does none of this. That is why the check that matters is application-layer reflection, not the certificate or the address.

Do not allowlist all of AWS

A tempting fix is to allowlist the whole Amazon network. That would also hide the 523 active Interactsh servers hosted on AWS EC2 in the same dataset. The answer is a structural check per host, not a network-wide exception.

CHAPTER 04

PHP 8.5, MySQL 3306 and Test Fixtures

IN SHORT

Application-layer checks clear the remaining 1,854 false positives: 127 PHP 8.5.10 websites, 1,592 MySQL servers and 135 test or prompt files. Each one fails a simple check that real C2 passes.

INSIDE THIS CHAPTER

Class 3: PHP 8.5.10 websites labelled ezXSS 14
Class 4: MySQL servers labelled Daybreak C2 15
Class 5: test fixtures flagged as live secrets 16
How the 5,576 secret findings split 17

04 PHP 8.5, MySQL 3306 and Test Fixtures

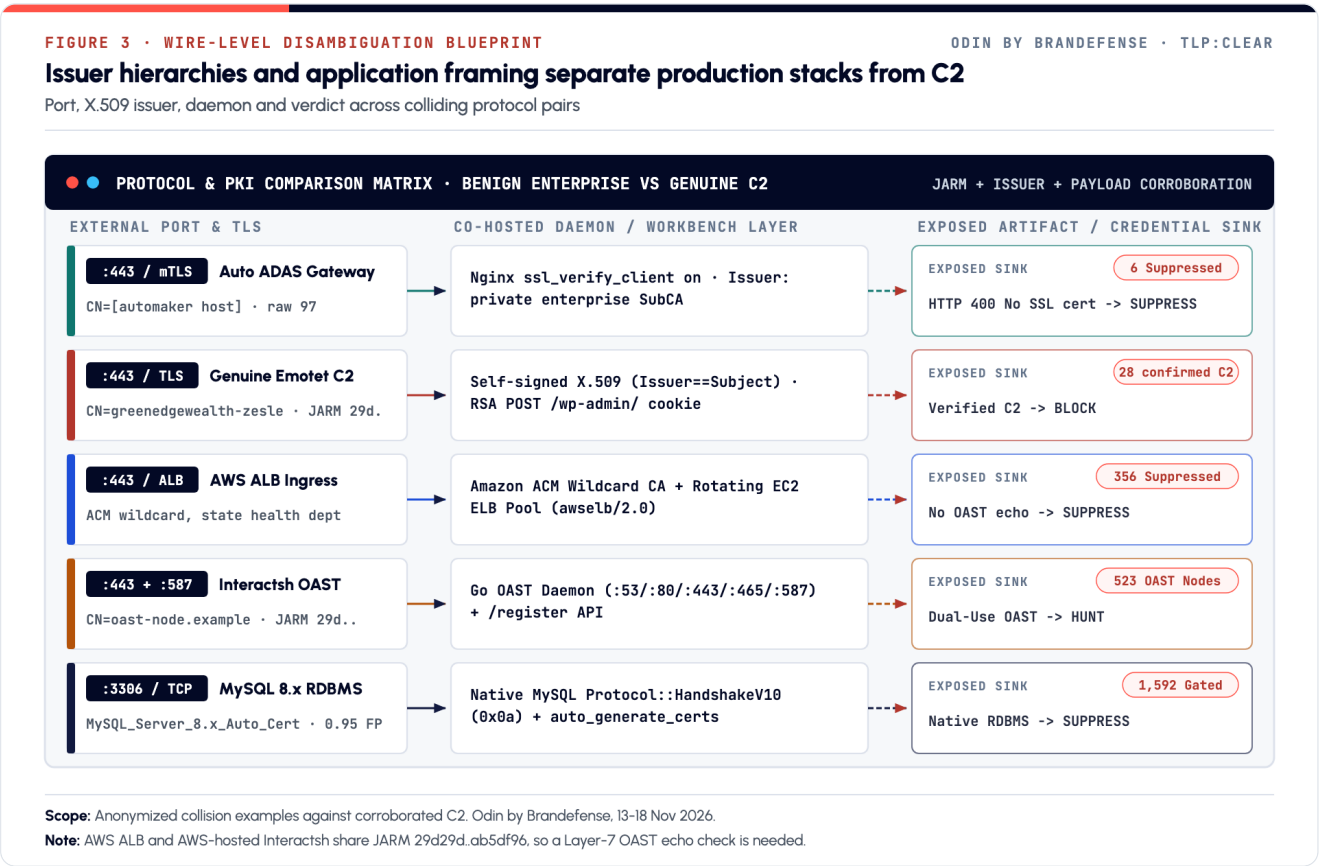


Figure 3: Issuer hierarchies and application framing separate production stacks from C2

Class 3: PHP 8.5.10 websites labelled ezXSS

A weak ezXSS rule fired on 127 endpoints across 101 unique IPv4 hosts. It scored 106 of them at 65.0 and 21 at 60.0. The hosts were commercial websites, event agencies, regional web portals and newsletter servers.

Hosting network	Rows
Amazon.com, Inc.	84
Hetzner Online GmbH	17
Contabo GmbH	6
Hangzhou Alibaba Advertising	4
Other cloud providers	16

The rows sit on ports 80, 443 and 8080.

ezXSS is an open-source blind cross-site scripting (XSS) collector. Many Docker deployments run modern PHP-FPM behind Nginx with `expose_php = On`. They answer with `X-Powered-By: PHP/8.x`, `HTTP/1.1 200 OK`, a `PHPSESSID` cookie and a strict `Content-Security-Policy` header.

When site owners upgraded ordinary CMS and newsletter servers to `PHP/8.5.10`, they emitted the same header mix. A header-only rule then labelled each upgraded server as suspected ezXSS. We call this runtime version drift: a legitimate upgrade moves benign servers into a rule's match range.

What a real ezXSS panel shows instead.

The two confirmed ezXSS servers in our indicator set expose fixed application markers:

- the operator login form at `/manage/login`
- a page title starting with `<title>ezXSS`
- the collector script `/assets/js/ezxss.js`, which harvests `document.cookie` and `localStorage`

They are 95.156.207[.]122:80 (`CN=xss.retr023[.]xyz`, M247 Europe) and 201.79.4[.]122:80 (`CN=a.fgeio[.]com`, DigitalOcean).

Class 4: MySQL servers labelled Daybreak C2

By volume, MySQL on port 3306 was the largest source of feed pollution. It accounts for 1,592 of the 2,081 C2 collisions (76.5%), each on a distinct IPv4 address. Every one of them carried the raw status "confirmed" at a score of 0.95 under the `daybreak` label.

Hosting network	MySQL collision hosts
Shenzhen Tencent Computer Systems	391
Hangzhou Alibaba Advertising	285
DigitalOcean, LLC	83
POWER LINE (HK)	63
Huawei Cloud	53

Two traits of a normal MySQL server explain the collision.

Trait one: the server speaks first.

Right after a TCP connection on port 3306, MySQL and MariaDB send `Protocol::HandshakeV10`. The packet starts with a 3-byte length, sequence byte 0x00 and protocol-version byte `0x0a`. A null-terminated version string follows, such as `8.0.45`, `5.7.44` or `9.1.0`. Then comes an authentication plugin name such as `caching_sha2_password` or `mysql_native_password`.

A port-agnostic C2 rule that matches a short, length-prefixed binary greeting plus an optional TLS upgrade will match this handshake.

Trait two: MySQL creates its own certificates.

Modern MySQL runs `auto_generate_certs` at startup. It creates a self-signed CA and a server certificate signed by that CA, with predictable subjects.

Auto-generated server certificate subject	Hosts
`MySQL_Server_5.7.44_Auto_Generated_Server_Certificate`	365
`MySQL_Server_8.0.45_Auto_Generated_Server_Certificate`	148
`MySQL_Server_8.0.46_Auto_Generated_Server_Certificate`	124

A rule that treats any auto-generated certificate over a binary greeting as C2 cannot work. Of 1,684 raw confirmed and suspected Daybreak rows, 1,592 were MySQL. That is a 94.5% false-positive rate among actionable rows, or 95.7% of the raw confirmed rows.

> **What real Daybreak C2 looked like.**
>
> Only 71 endpoints, none on port 3306, held up as confirmed Daybreak C2, plus 17 on dynamic ISP networks. Examples in our indicator set are 188.68.60[.]39:20000 (netcup, also active on 80 and 443), 46.105.117[.]146:443 (OVH, self-signed `CN=radeau`) and 103.137.250[.]153:1501 (CLODO, `CN=keitaro.2` on 1500 and 1501). The name "Daybreak" is the feed's label for this framework.

Class 5: test fixtures flagged as live secrets

Regex-only secret and callback scanners have the same blind spot when they read exposed developer directories. Of 5,576 secret findings, 135 were false positives: 41 critical, 15 high and 79 medium. They sat on 28 unique IPs across 29 open-directory endpoints.

Three patterns drove them.

Pattern	Example	Why it is not a leak
AI skill docs and public prompt archives	Mirrored system-prompt collections that document `.cursor` and `.claude` prompts; skill reference docs under `.hermes/skills/` with a tutorial placeholder password	Path names trigger `.cursor`, `.claude` and `.hermes` rules, but the files hold no live `sessionId` or OAuth state
Unit tests with vendor example keys	`*_test.go` and `.test.js` files with AWS's documented example key `AKIAIOSFODNN7EXAMPLE`, mock Slack `xoxb-fake-...` tokens and an RFC 1918 test callback `http://192.168.1.100:8080/api` (60 fixture hits)	Public documentation values and private test addresses
Library self-tests	PyCryptodome's `Crypto/SelfTest/loader.py` (8,518 bytes) in a Python 3.12 virtualenv, tagged as a C2 payload	The unit-test loader calls cipher routines, nothing more

The same AWS example key also appeared in a Metasploit wordlist file on another host. That is why known example values need their own check, alongside path filtering.

Path filtering must not hide real leaks. A live credential placed under a `/tests/` folder is still live. Live session paths should always be checked first.

How the 5,576 secret findings split

Finding type	Records	Share
Hard-coded adversary C2 or callback addresses	2,175	39.0%
Embedded C2 or callback strings awaiting corroboration	1,663	29.8%
Exposed third-party corporate or application secrets	797	14.3%
Credentials staged on adversary workbenches	675	12.1%
Open-source test fixtures (false positives)	135	2.4%
Live AI agent session leaks (`.claude` , `.cursor` , `.codex` , `.openclaw` , `.hermes`)	131	2.3%

The bottom two rows are the point. The 135 fixtures slightly outnumber the 131 live AI agent session leaks. Without path context, the fixtures compete with real exposures for analyst time.

05

CHAPTER 05

How Much of a Raw Feed Is Safe to Block

IN SHORT

Of 25,482 raw indicators, 17.4% are confirmed C2 and 15.4% are fit for a block list. Before any checks, false-positive rates in four frameworks ran from 9.7% to 94.5%.

INSIDE THIS CHAPTER

Where the 25,482 indicators land 19

False-positive rates before any checks 20

Residential and mobile networks need short lifetimes 20

What separates real C2 from a collision 21

05 How Much of a Raw Feed Is Safe to Block

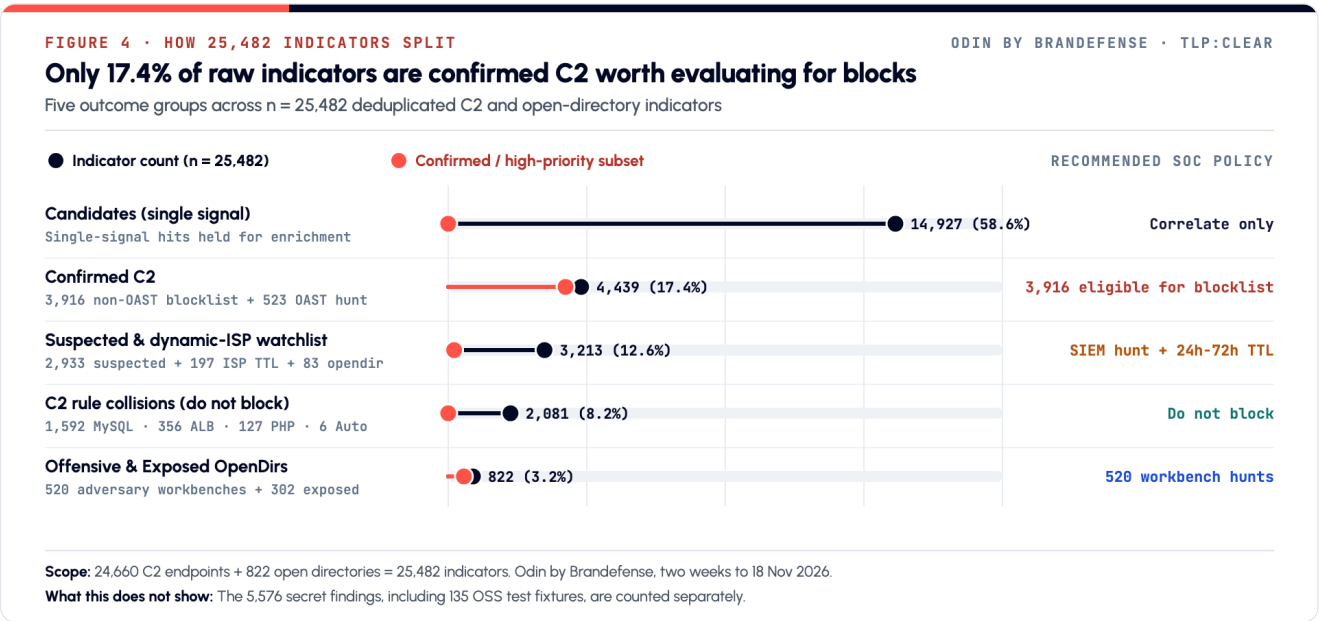


Figure 4: Only 17.4% of raw indicators are confirmed C2 worth evaluating for blocks

Where the 25,482 indicators land

The dataset holds 24,660 C2 endpoints and 822 open directories. Most of it is not safe to block.

Outcome	Indicators	Share	Recommended handling
Candidates with a single signal	14,927	58.6%	Correlate only; never block
Confirmed C2	4,439	17.4%	3,916 to block; 523 dual-use Interactsh servers to hunting
Suspected C2 and short-life watchlist entries	3,213	12.6%	SIEM watchlist; 24 to 72 hours for subscriber ISPs
C2 rule collisions on benign servers	2,081	8.2%	Do not block; keep as regression tests
Offensive and exposed open directories	822	3.2%	520 adversary workbenches to hunt; 302 exposed directories to watch

The 3,916 block-ready indicators (15.4%) are confirmed C2 that a separate signal corroborates. The 523 Interactsh servers (2.1%) are real OAST servers, but OAST is a dual-use testing tool. They go to hunting unless independent hostile-callback evidence appears.

The 3,213 watchlist entries are 2,933 suspected C2, 197 dynamic ISP hits and 83 candidates that share a host with an open directory.

False-positive rates before any checks

The table below shows how collisions distort four raw framework feeds. "Raw actionable" counts rows labelled confirmed or suspected before any check runs.

Framework label	Total rows	Raw actionable	Collisions	Actionable false-positive rate	Confirmed C2 after checks
`daybreak`	1,686	1,684	1,592	94.5%	71
`ezxss`	252	177	127	71.8%	2
`interactsh`	1,600	994	356	35.8%	523 (dual-use, to hunting)
`emotet`	62	62	6	9.7%	28

Two lessons follow. A feed that ships raw Daybreak or Interactsh detections delivers 94.5% MySQL handshakes or 35.8% AWS load balancers. And even after collisions are removed, the 523 Interactsh servers are testing infrastructure, so they belong in hunting rather than on a block list.

Residential and mobile networks need short lifetimes

A permanent IP block also fails when the C2 host sits on a dynamic subscriber pool. In our data, 197 endpoints did: 112 confirmed and 85 suspected, on 161 unique IPs. The frameworks were Mozi, Gh0st RAT, TacticalRMM, Brute Ratel C4 and Daybreak.

The networks were Chinanet, China Unicom, Deutsche Telekom AG, Chunghwa Telecom, Globofiber Telecom and Comcast. Carrier DHCP leases vary, and a subscriber address can pass to a new household without notice.

Example (address withheld)	Network	Status	Watchlist lifetime
TacticalRMM listener on port 1024	Globofiber Telecom (BR)	Confirmed	72 hours
Mozi botnet peer on port 443	Chinanet (CN)	Suspected	24 hours
Gh0st RAT listener on port 8443	Chunghwa Telecom (TW)	Suspected	24 hours
Brute Ratel C4 on port 8081	China Unicom (CN)	Suspected	24 hours

These addresses are not published, because they may belong to unrelated households by the time a reader sees them.

What separates real C2 from a collision

Every collision in this report fails at least one of the checks below, and every confirmed C2 server passes them. No single check is enough. A server is treated as confirmed C2 only when several independent signals agree.

Check	Real C2 tends to show	A collision tends to show
Certificate chain	Self-signed certificate, issuer equals subject	A private enterprise sub-CA or an Amazon ACM chain
TLS error behaviour	Short 400 or 404 to a payload-less POST	"No required SSL certificate was sent": a mutual-TLS gateway
Application reflection	Interactsh reflects the correlation ID and answers `/register` on several ports	A default `awselb/2.0` answer with no reflection
Application markers	ezXSS `/manage/login`, `<title>ezXSS`, `ezxss.js` collector	A plain PHP version header and a session cookie
Protocol framing	A C2 check-in on a non-database port	MySQL greeting byte `0x0a` and an auth plugin name on 3306
Network type	Stable hosting or VPS network	Residential or mobile subscriber pool: watch briefly, do not block
File context	Live session state in an agent folder	Test folders, skill docs, prompt archives, known example keys

A positive hostile signal always wins. If a server shows a benign stack and also a real C2 beacon, panel or payload, it goes to analyst review rather than being cleared.

06

CHAPTER 06

What Defenders Should Do

IN SHORT

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

06 What Defenders Should Do

Within 24 hours

- [ ] Stop SOAR and firewall playbooks from auto-blocking on a single signal or a score alone.
- [ ] Remove any port-3306 Daybreak, header-only ezXSS, JARM-only Interactsh and HTTP-400-only Emotet indicators from automated block jobs.
- [ ] Block the eight confirmed C2 endpoints listed in Chapter 8.

Within 72 hours

- [ ] Restrict auto-blocking to confirmed C2 on stable hosting that carries a second, framework-specific signal.
- [ ] Route Interactsh and other OAST servers to a 7-day hunting queue unless hostile callbacks are seen.
- [ ] Put C2 hits on residential and mobile networks on watchlists that expire after 24 hours (suspected) or 72 hours (confirmed).

Within 7 to 30 days

- [ ] Add certificate issuer checks and mTLS rejection checks to feed ingestion.
- [ ] Add MySQL handshake detection and ezXSS panel markers to C2 enrichment.
- [ ] In secret scanning, check live `.claude`, `.cursor`, `.codex`, `.openclaw` and `.hermes` session paths first. Then filter `/tests/`, skill docs, prompt archives and known example keys.
- [ ] Load the six do-not-block placeholder endpoints from the Sentinel rule into staging, plus the three named fixtures in Chapter 8. Confirm that none of them raises a block or an alert.
- [ ] Never fix a collision by allowlisting a whole cloud network. Add exact, verified partner PKI domains instead.

07

CHAPTER 07

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 other rules work 25

07 Detection Rules

The full rules are in Appendix B. Each file pairs a positive C2 signal with an explicit exclusion for the collision classes in this report.

Rule file	What it catches
`odin_c2_collision_safe.rules`	Suricata 7: eight signatures for Emotet, Interactsh, ezXSS and Daybreak, plus a MySQL guardrail that keeps port 3306 handshakes out of C2 alerts
`sentinel_ti_feed_collision_guardrail.kql`	Microsoft Sentinel and Defender XDR: filters a threat-intel feed for the collision classes and short ISP lifetimes, then matches it against outbound connections
`sigma_soar_blocklist_enterprise_pki_safeguard.yml`	Sigma (3 rules): uncalibrated SOAR auto-block jobs, scripts connecting to confirmed C2, and suspicious reads of live AI agent credential stores
`odin_benign_fixture_vs_live_c2_disambiguator.yar`	YARA (4 rules): Emotet, Interactsh, ezXSS and Daybreak C2 content, and live agent transcripts or Sliver operator logs, with fixture exclusions

How the Suricata signatures work

SID	Looks for	Why
9513001	The Emotet-associated certificate issuer `CN=greenedgewealth-zesle`, excluding sub-CA and Amazon RSA issuers	Hunt lead; pair with HTTP or endpoint evidence
9513002	An Emotet Epoch 4/5 style encoded POST to `/wp-admin/` with a `PHPSESSID` cookie	Hunt lead; confirm with the process that sent it, such as `rundll32.exe` or `regsvr32.exe`
9513003	The Interactsh HTML banner in a 200 response	Finds OAST servers; load balancers do not serve this banner
9513004	Interactsh `/poll?id=` requests with an `Authorization` header	Finds internal hosts polling an OAST server
9513005	The ezXSS collector script `/assets/js/ezxss.js` with `document.cookie` and `localStorage`	Real ezXSS content, safe against PHP 8.5 header drift
9513006	`<title>ezXSS` with `/manage/login`	The ezXSS operator panel
9513007	A Daybreak check-in POST to `/api/v1/daybreak/` with an `X-Daybreak-Node:` header	Real Daybreak C2, skipped on flows marked as MySQL
9513008	A MySQL handshake from port 3306 (byte `0x0a` and an auth plugin name)	Sets a flag without alerting, so later rules can skip the flow

The Interactsh signatures (9513003 and 9513004) find a dual-use tool. Treat their hits as hunting leads. To exclude an authorized internal OAST server, add a local `pass` rule for its exact /32 address rather than disabling the signature.

How the other rules work

The KQL query reads the `ThreatIntelIndicators` table over 14 days and keeps only active, non-revoked IPv4 indicators. It then drops the four collision patterns. These are Daybreak on 3306 with a MySQL handshake, low-score ezXSS with a PHP 8.5 header, mTLS rejections from an exact partner hostname, and Interactsh on an AWS load balancer.

Each drop applies only when the feed carries no hostile evidence, and a missing enrichment field never drops a row. The query then enforces a lifetime per indicator type, including 72 hours for confirmed and 24 hours for suspected subscriber ISP hits. Confidence cut-offs are parameters; set them to your feed's own scale. Finally it joins the result to `DeviceNetworkEvents` over 7 days.

The first Sigma rule fires when a SOAR job pushes a firewall or EDR block for a risky indicator. That covers port-3306 Daybreak, suspected Amazon-hosted hits, candidates, known collisions, dynamic ISP hits and dual-use OAST servers. The second fires when a scripting or system process connects to confirmed C2, with partner mTLS and load-balancer exclusions. The third fires when an archive, transfer or scripting process reads a live AI agent credential store, while ignoring static skill and prompt files.

Each YARA rule needs complete positive C2 or live-session evidence before it fires. Fixture strings such as test paths and example keys act as exclusions, never as triggers.

The triage logic takes one feed record and returns an action. Its routing follows the checks in Chapter 5.

What the record shows	Action
Confirmed C2 with a framework-specific hostile signal, on stable hosting	Block for 14 days
Uncorroborated self-signed C2, or a benign stack that also shows a hostile signal	Watchlist for 7 days, analyst review
Interactsh or other OAST server without hostile callbacks	Hunt for 7 days
C2 hit on a residential or mobile subscriber network	Watchlist for 24 or 72 hours
A verified benign stack (mTLS gateway, load balancer, PHP 8.5 header, MySQL handshake) and no hostile signal	Do not block
A test fixture or known example key, with no live session state	Do not alert

The rules map to MITRE ATT&CK T1071.001 (Application Layer Protocol: Web Protocols) and T1583.006 (Acquire Infrastructure: Web Services).

CHAPTER 08

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Confirmed C2 endpoints to block 28

Hunt and regression records 28

08 Indicators

Appendix A lists the eight confirmed C2 endpoints. The other records in this chapter are described but not published.

Type	Count
Confirmed C2 endpoints to block (IP and port)	8
Exposed Sliver operator workbench, hunt only	1 (address withheld)
Interactsh OAST servers, hunt only	3 (addresses withheld)
Dynamic subscriber ISP examples, short watchlist	4 (addresses withheld)
Secret-finding digests (withheld: they derive from exposed user data)	3
Do-not-block regression samples	9 (6 placeholder endpoints in the Sentinel rule, 3 named fixtures)

Confirmed C2 endpoints to block

Endpoint	Label	Hosting (country)	Key fingerprint
64.227.189[.]225:443	Emotet	DigitalOcean (IN)	Self-signed `CN=greenedgewealth-zesle`, JARM `29d29d15d..579afa`
128.199.131[.]70:443	Emotet	DigitalOcean (SG)	Self-signed `CN=serv-01.nexgeninter[.]net`, JARM `29d29d15d..579afa`
167.71.229[.]25:443	Emotet	DigitalOcean (IN)	Self-signed `CN=Callsign`, JARM `29d29d15d..579afa`
95.156.207[.]122:80	ezXSS	M247 Europe (DE)	`CN=xss.retr023[.]xyz`, collector panel
201.79.4[.]122:80	ezXSS	DigitalOcean (DE)	`CN=a.fgeio[.]com`, collector panel
188.68.60[.]39:20000	Daybreak	netcup (DE)	Also active on 80 and 443
46.105.117[.]146:443	Daybreak	OVH (FR)	Self-signed `CN=radeau`
103.137.250[.]153:1501	Daybreak	CLODO (NL)	`CN=keitaro.2` on 1500 and 1501

Cloud VPS addresses can be reassigned after a takedown. Re-check that the listener is still active before you extend a block past 14 days.

Hunt and regression records

Three secret-finding digests trace a Sliver stager address and two live AI agent credential exposures. They are withheld, because they derive from exposed user data.

The nine do-not-block samples cover all five collision classes. Six are placeholder addresses from the non-routable documentation ranges, one per benign stack, listed as `CollisionCanaryIps` in the Sentinel rule. Three are known fixtures, named rather than hashed: the AWS documentation example key, a skill-doc placeholder and the PyCryptodome self-test loader. Use them to test that your ingestion filters clear every collision class.

Withheld: the automaker, state agency and cloud customers behind the collisions, all 1,958 benign collision hosts, the Sliver workbench, the three Interactsh servers, the dynamic ISP addresses, and the hosts behind the secret findings.

09

CHAPTER 09

Scope and Limits

IN SHORT

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

09 Scope and Limits

- **Window.** The findings cover a two-week observation window ending 18 November 2026. The collisions were seen between 13 and 18 November 2026.
- **External view only.** All findings come from handshakes, certificates, HTTP headers and open directories visible from the internet. No customer logs were used.
- **Modeled impact.** The service risks of a false block are modeled from the protocols and owners involved. We did not see any actual block or outage.
- **Feed labels are not attribution.** Framework names such as Daybreak and Emotet come from detection feeds. A collision row says nothing about the benign owner's intent.
- **Dual-use is not hostile.** The 523 Interactsh servers run a legitimate testing tool. Some may belong to authorized testers.
- **Counts, not trends.** The data supports counts in the window, not growth or decline over time.
- **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.

10

CHAPTER 10

Sources

IN SHORT

This chapter lists the public references used in the report.

10 Sources

1. Salesforce. JARM active TLS server fingerprinting. https://github.com/salesforce/jarm (verified 18 November 2026)
2. ProjectDiscovery. Interactsh OAST server. https://github.com/projectdiscovery/interactsh (verified 18 November 2026)
3. Nginx. `ngx_http_ssl_module`, `ssl_verify_client`. https://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_verify_client (verified 18 November 2026)
4. IETF. RFC 8446, The Transport Layer Security (TLS) Protocol Version 1.3. https://datatracker.ietf.org/doc/html/rfc8446 (verified 18 November 2026)
5. Amazon Web Services. Create an HTTPS listener for your Application Load Balancer. https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html (verified 18 November 2026)
6. The PHP Group. Core `php.ini` directives, `expose_php`. https://www.php.net/manual/en/ini.core.php#ini.expose-php (verified 18 November 2026)
7. Oracle MySQL. `Protocol::HandshakeV10`. https://dev.mysql.com/doc/dev/mysql-server/latest/page_protocol_connection_phase_packets_protocol_handshake_v10.html (verified 18 November 2026)
8. Oracle MySQL. Creating SSL and RSA certificates and keys using MySQL. https://dev.mysql.com/doc/refman/8.4/en/creating-ssl-rsa-files-using-mysql.html (verified 18 November 2026)
9. NIST. SP 800-150, Guide to Cyber Threat Information Sharing. https://csrc.nist.gov/pubs/sp/800/150/final (verified 18 November 2026)
10. MITRE ATT&CK. T1071.001 Application Layer Protocol: Web Protocols. https://attack.mitre.org/techniques/T1071/001/ (verified 18 November 2026)
11. MITRE ATT&CK. T1583.006 Acquire Infrastructure: Web Services. https://attack.mitre.org/techniques/T1583/006/ (verified 18 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 (8) 35

B. Detection rules (4 files) 36

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
R13-IOC-01	IPv4	64.227.189[.]225	443	emotet	BLOCK
R13-IOC-02	IPv4	128.199.131[.]70	443	emotet	BLOCK
R13-IOC-03	IPv4	167.71.229[.]25	443	emotet	BLOCK
R13-IOC-04	IPv4	95.156.207[.]122	80	ezxss	BLOCK
R13-IOC-05	IPv4	201.79.4[.]122	80	ezxss	BLOCK
R13-IOC-06	IPv4	188.68.60[.]39	20000	daybreak	BLOCK
R13-IOC-07	IPv4	46.105.117[.]146	443	daybreak	BLOCK
R13-IOC-08	IPv4	103.137.250[.]153	1501	daybreak	BLOCK

B Detection Rules

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

odin_benign_fixture_vs_live_c2_disambiguator.yar

YARA

```
/*
   Odin by Brandefense - Multi-Signal C2 & OSS Fixture Disambiguation YARA Suite (Report #13)
   Classification: TLP:CLEAR | Date: 2026-11-18
   Evaluated against 12 normalized protocol/artifact specimens (6/6 positive signal hits, 0/6 collision FPs).
*/

rule ODIN_Emotet_SelfSigned_TLS_And_Post_C2
{
    meta:
        author = "Odin by Brandefense"
        classification = "TLP:CLEAR"
        description = "Detects self-signed Emotet TLS/HTTP POST C2 profiles (matching certificate subject and
issuer plus the POST profile) while excluding enterprise SubCA ADAS mTLS HTTP 400 gateways"
        report_ref = "Monograph-13"
    strings:
        $c2_post = "POST /wp-admin/" ascii
        $c2_form = "Content-Type: application/x-www-form-urlencoded" ascii
        $c2_cookie = "Cookie: PHPSESSID=" ascii
        $c2_subj = "Subject: CN=greenedgewealth-zesle" ascii
        $c2_iss = "Issuer: CN=greenedgewealth-zesle" ascii
        $c2_jarm = "29d29d15d29d29d00042d42d0000008fe5654c9239cdb4052d3ab65a579afa" ascii

        $fp_mtls_400 = "No required SSL certificate was sent" ascii nocase
        $fp_auto_ca = "Identity SubCA RSA" ascii nocase
        $fp_auto_dom = ".auto-oem.example" ascii nocase
    condition:
        filesize < 2MB and
        (
            ($c2_post and $c2_form and $c2_cookie and $c2_subj and $c2_iss and $c2_jarm) or
            (($c2_post and $c2_form and $c2_subj and $c2_iss) and not any of ($fp_*))
        )
}

rule ODIN_Interactsh_OAST_Reflection_Server
{
    meta:
        author = "Odin by Brandefense"
        classification = "TLP:CLEAR"
        description = "Detects active dual-use Interactsh OAST reflection responses for suspected-tier hunting
while excluding AWS ALB ACM wildcard endpoints"
        report_ref = "Monograph-13"
    strings:
        $oast_h1 = "<h1> Interactsh Server </h1>" ascii
        $oast_hdr = "X-Interactsh-Version:" ascii nocase
        $oast_srv = "Server: interactsh" ascii nocase
        $oast_api1 = "/register" ascii
        $oast_api2 = "/poll" ascii

        $fp_awselb = "Server: awselb/" ascii nocase
        $fp_amazon_ca = "CN=Amazon RSA" ascii
        $fp_state_health = "state-health.gov.example" ascii nocase
    condition:
        filesize < 2MB and
        (
            ($oast_h1 and $oast_hdr and $oast_srv and $oast_api1 and $oast_api2) or
            ((($oast_h1 and ($oast_hdr or $oast_srv)) or ($oast_h1 and $oast_api1 and $oast_api2)) and not any of
($fp_*))
        )
}
```

```
rule ODIN_ezXSS_Callback_And_Daybreak_C2
{
    meta:
        author = "Odin by Brandefense"
        classification = "TLP:CLEAR"
        description = "Detects ezXSS collector DOM panels and Daybreak C2 beacons while excluding PHP 8.5 headers
and MySQL 3306"
        report_ref = "Monograph-13"
    strings:
        $ez_title = "<title>ezXSS" ascii nocase
        $ez_js = "/ezxss.js" ascii nocase
        $ez_login = "/manage/login" ascii
        $ez_dom1 = "document.cookie" ascii
        $ez_dom2 = "localStorage" ascii

        $db_uri = "/api/v1/daybreak/" ascii
        $db_hdr = "X-Daybreak-Node:" ascii nocase
        $db_magic = "DBRK_BEACON" ascii

        $fp_php85 = "X-Powered-By: PHP/8.5." ascii nocase
        $fp_mysql_cert = "Auto_Generated_Server_Certificate" ascii
        $fp_mysql_auth = "mysql_native_password" ascii
    condition:
        filesize < 2MB and
        (
            (
                (($ez_title and $ez_js) and ($ez_login or ($ez_dom1 and $ez_dom2))) or
                ((($ez_title or $ez_js) and ($ez_login or ($ez_dom1 and $ez_dom2))) and not $fp_php85)
            ) or
            (
                ($db_magic and $db_uri and $db_hdr) or
                (($db_magic and ($db_uri or $db_hdr)) and not ($fp_mysql_cert or $fp_mysql_auth))
            )
        )
}

rule ODIN_Live_Agent_Transcript_And_Workbench_C2_Leak
{
    meta:
        author = "Odin by Brandefense"
        classification = "TLP:CLEAR"
        description = "Detects live autonomous agent JSONL transcripts and Sliver C2 operator logs while
suppressing OSS test and public prompt-archive fixtures"
        report_ref = "Monograph-13"
    strings:
        $ag_s1 = "\"sessionId\":\"" ascii
        $ag_s2 = "\"tool_use\"" ascii
        $ag_s3 = "/.claude/projects/" ascii
        $ag_s4 = "/.openclaw/identity/device.json" ascii
        $ag_live = "\"parentUuid\":\"" ascii

        $sl_1 = "/.sliver/logs/clients/" ascii
        $sl_2 = "sliver-client" ascii
        $sl_3 = "execute-assembly" ascii

        $fp_aws_ex = "AKIAIOSFODNN7EXAMPLE" ascii
        $fp_unsloth = "unsloth2024" ascii
        $fp_prompt1 = "/ai-prompt-archive-fixture/" ascii nocase
        $fp_prompt2 = "Public AI System-Prompt Reference Archive" ascii nocase
        $fp_skills = "/.hermes/skills/" ascii
        $fp_test1 = ".test.js" ascii
        $fp_test2 = "_test.go" ascii
    condition:
        filesize < 2MB and
        (
            ($ag_s1 and $ag_s2 and $ag_live and ($ag_s3 or $ag_s4)) or
            ($sl_1 and $sl_2 and $sl_3) or
            ((($ag_s1 and $ag_s2) and ($ag_s3 or $ag_s4)) and not any of ($fp_*))
        )
}
```

odin_c2_collision_safe.rules

SURICATA

```
# Odin by Brandefense - Collision-Safe Multi-Signal C2 Suricata 7.x Suite (Report #13)
# Classification: TLP:CLEAR | SIDs: 9513001 - 9513008
# Purpose: Detect genuine Emotet, ezXSS, and Daybreak C2 traffic plus dual-use Interactsh
# OAST callbacks while explicitly suppressing enterprise SubCA ADAS mTLS (:443), AWS ALB
# wildcard rotation, PHP 8.5.10 header drift, and native MySQL TCP/3306 HandshakeV10 greetings.

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [Suspected Hunt] Emotet-Associated TLS Server Certificate
Issuer CN (Excludes Enterprise SubCA mTLS, Corroborate With HTTP Or Endpoint Evidence)";
flow:established,to_client; tls.cert_subject; content:"CN="; tls.cert_issuer; content:"CN=greenedgewealth-zesle";
fast_pattern; content:!"Identity SubCA"; content:!"auto-oem.example"; content:!"SubCA"; content:!"Amazon RSA";
classtype:trojan-activity; sid:9513001; rev:2;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [Suspected Hunt] Emotet Epoch4/5 Encoded HTTP POST
Cookie Beacon (Correlate With Endpoint Process)"; flow:established,to_server; http.method; content:"POST";
http.uri; content:"/wp-admin/"; fast_pattern; http.header; content:"Content-Type: application/x-www-form-
urlencoded"; http.cookie; content:"PHPSESSID="; bsize:>12; http.request_body; content:"="; pcre:"/^[A-Za-z0-9_]
{4,24}=[A-Za-z0-9+%\/]{32,}/"; classtype:trojan-activity; sid:9513002; rev:3;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [Suspected Hunt] Dual-Use Interactsh OAST HTML
Reflection Banner"; flow:established,to_client; http.stat_code; content:"200"; http.response_body; content:"<h1>
Interactsh Server </h1>"; fast_pattern; classtype:command-and-control; sid:9513003; rev:2;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [Suspected Hunt] Dual-Use Interactsh OAST Polling and
Registration API"; flow:established,to_server; http.uri; content:"/poll?id="; fast_pattern; http.header_names;
content:"Authorization"; classtype:command-and-control; sid:9513004; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [Confirmed] ezXSS Blind-XSS DOM Collector Script
Delivery (PHP 8.5 Safe)"; flow:established,to_client; http.stat_code; content:"200"; http.response_body;
content:"/assets/js/ezxss.js"; fast_pattern; content:"document.cookie"; content:"localStorage"; classtype:web-
application-attack; sid:9513005; rev:1;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [Confirmed] ezXSS Operator Management Panel
Authentication Page"; flow:established,to_client; http.stat_code; content:"200"; http.response_body;
content:"<title>ezXSS"; fast_pattern; content:"/manage/login"; classtype:command-and-control; sid:9513006; rev:
1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [Confirmed] Daybreak C2 Beacon Check-In Frame
(Suppressed Only On Flows With A Verified MySQL Handshake)"; flow:established,to_server;
flowbits:isnotset,odin.benign_mysql_3306; http.method; content:"POST"; http.uri; content:"/api/v1/daybreak/";
fast_pattern; http.header; content:"X-Daybreak-Node:"; classtype:command-and-control; sid:9513007; rev:2;)

alert tcp $EXTERNAL_NET 3306 -> $HOME_NET any (msg:"ODIN [Suppression] Native MySQL HandshakeV10 Port 3306
Flowbit Suppression"; flow:established,to_client; dsize:>32; content:"|0a|"; offset:4; depth:1;
content:"_password"; fast_pattern; pcre:"/(?:mysql_native|caching_sha2|sha256)_password\x00/";
flowbits:set,odin.benign_mysql_3306; flowbits:noalert; classtype:protocol-command-decode; sid:9513008; rev:3;)
```

sentinel_ti_feed_collision_guardrail.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Threat Intelligence Feed Collision Guardrail (Report #13)
// Platform: Microsoft Sentinel (ThreatIntelIndicators) / Microsoft Defender XDR (DeviceNetworkEvents)
// Classification: TLP:CLEAR
// Drops four known collision patterns from a C2 indicator feed (MySQL on 3306 labelled Daybreak, PHP 8.5
// header drift labelled ezXSS, partner mTLS rejections, Interactsh labels on AWS load balancers), applies
// short lifetimes to subscriber ISP addresses, then matches the remaining indicators to outbound connections.
// Every enrichment column below is optional. When a column is missing, the matching collision check does not
// drop the indicator, so nothing is silently discarded. Set the confidence values to your own feed's scale.
// Loads in Kusto.Language; not replayed against live data, so test in your own environment first.
let Lookback = 7d;
let MinBlockConfidence = 85.0;      // your feed's "confirmed" level
let MaxLowConfidence = 66.0;        // your feed's "low confidence" ceiling
let ConfirmedTtl = 14d;
let DefaultTtl = 7d;
let SubscriberIspConfirmedTtl = 72h;
let SubscriberIspSuspectedTtl = 24h;
```

```
// Exact partner hostnames only; add your own. No suffix or wildcard matching.
let PartnerMtlsHosts = dynamic(["adas-mtls.auto-oem.example", "telematics.auto-oem.example"]);
let LocalAlbHosts = dynamic(["alb-tenant-a.example", "alb-tenant-b.example"]);
// Do-not-block placeholder endpoints (documentation ranges) for staging tests.
let CollisionCanaryIps = dynamic(["192.0.2.17", "192.0.2.18", "198.51.100.19", "198.51.100.20", "203.0.113.21",
"203.0.113.22"]);
let SubscriberIspKeywords = dynamic([
    "telecom", "telekom", "chinanet", "china unicom", "china mobile",
    "chunghwa", "globofiber", "comcast", "verizon", "vodafone"
]);
let Feed =
    ThreatIntelIndicators
    | where TimeGenerated >= ago(14d) and isnotempty(Id)
    | summarize arg_max(TimeGenerated, *) by Id
    | extend DataObj = parse_json(tostring(Data)), NetworkIP = trim(" ", tostring(ObservableValue))
    | where IsActive == true and not(coalesce(IsDeleted, false)) and not(coalesce(tobool(DataObj.revoked),
false))
    | where isnotempty(ObservableValue) and (isempty(ObservableKey) or ObservableKey in ("ipv4-addr:value",
"network-traffic:dst_ref.value"))
    | where NetworkIP matches regex @"^(?:\d{1,3}\.){3}\d{1,3}$" and NetworkIP !in (CollisionCanaryIps)
    | extend Extra = iff(isnotnull(DataObj.additional_information), DataObj.additional_information, DataObj)
    | extend FeedPort = toint(coalesce(Extra.network_port, Extra.port, DataObj.network_port, DataObj.port)),
             Label = tolower(coalesce(tostring(Extra.framework), tostring(Extra.threat_type),
tostring(DataObj.name), tostring(Tags))),
             RawConfidence = coalesce(todouble(Confidence), todouble(Extra.confidence), 0.0),
             AsnOrg = tolower(coalesce(tostring(Extra.asn_name), tostring(Extra.asn_org), "")),
             CertSubject = replace_regex(tolower(coalesce(tostring(Extra.cert_subject_cn), "")), @"^(?:cn=|
\*\.)", ""),
             CertIssuer = tolower(coalesce(tostring(Extra.cert_issuer_cn), "")),
             PoweredBy = tolower(tostring(Extra.http_powered_by)),
             ServerHeader = tolower(tostring(Extra.http_server)),
             MysqlGreeting = coalesce(tobool(Extra.has_mysql_handshake_v10), false),
             MtlsRejection = coalesce(tobool(Extra.has_mtls_cert_rejection_400), false),
             HostileEvidence = coalesce(tobool(Extra.has_verified_c2_beacon_or_panel), false),
             LastObserved = coalesce(todatetime(Extra.last_seen_utc), todatetime(Extra.observed_at_utc),
TimeGenerated)
    | extend Conf = iff(RawConfidence > 0.0 and RawConfidence < 1.0, RawConfidence * 100.0, RawConfidence)
    // Collision 1: Daybreak label on a server that answers with a MySQL greeting on 3306
    | where not(Label contains "daybreak" and FeedPort == 3306 and MysqlGreeting and not(HostileEvidence))
    // Collision 2: low-confidence ezXSS label on a PHP 8.5 header
    | where not(Label contains "ezxss" and Conf <= MaxLowConfidence and PoweredBy contains "php/8.5" and
not(HostileEvidence))
    // Collision 3: partner mTLS gateway that rejects clients without a certificate
    | where not(MtlsRejection and CertSubject in (PartnerMtlsHosts) and not(HostileEvidence))
    // Collision 4: Interactsh label on an AWS load balancer (awselb server header, Amazon RSA issuer)
    | where not(Label contains "interactsh" and ServerHeader startswith "awselb/" and CertIssuer contains "amazon
rsa" and not(HostileEvidence))
    // Lifetimes: short for subscriber ISP addresses, longer for confirmed C2
    | extend IsSubscriberIsp = AsnOrg has_any (SubscriberIspKeywords)
    | extend Ttl = case(IsSubscriberIsp and Conf >= MinBlockConfidence, SubscriberIspConfirmedTtl,
                        IsSubscriberIsp, SubscriberIspSuspectedTtl,
                        Conf >= MinBlockConfidence, ConfirmedTtl,
                        DefaultTtl)
    | where LastObserved >= ago(Ttl)
    | where isnull(ValidUntil) or ValidUntil > now()
    | extend Handling = case(IsSubscriberIsp, "Watchlist_Subscriber_ISP",
                             Label contains "interactsh", "Hunt_Dual_Use_OAST",
                             Conf >= MinBlockConfidence and HostileEvidence, "Block_Confirmed_C2",  // a score
alone never yields a block
                             Conf >= MinBlockConfidence, "Review_High_Score_No_Corroboration",
                             "Hunt_Suspected_C2")
    | summarize arg_max(TimeGenerated, *) by NetworkIP, FeedPort;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess" and RemoteIPType == "Public"
| where RemoteIP !in (CollisionCanaryIps)
| extend CleanRemoteUrl = tolower(trim(" ", tostring(RemoteUrl)))
| extend ParsedHost = tostring(parse_url(iff(CleanRemoteUrl contains "://", CleanRemoteUrl, strcat("https://",
CleanRemoteUrl))).Host)
| extend RemoteHost = iff(isnotempty(ParsedHost), ParsedHost, CleanRemoteUrl)
| join kind=inner Feed on $left.RemoteIP == $right.NetworkIP
| where isnull(FeedPort) or RemotePort == FeedPort
// Partner mTLS and local load-balancer hosts are excluded by exact hostname, unless the feed carries hostile
```

```
evidence
| where HostileEvidence or isempty(RemoteHost) or not(RemoteHost in (PartnerMtlsHosts) or RemoteHost in
(LocalAlbHosts))
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), HitCount = count(), Processes =
make_set(InitiatingProcessFileName, 5), ObservedUrls = make_set(CleanRemoteUrl, 5) by DeviceName, RemoteIP,
RemotePort, Label, Conf, Handling
| order by Conf desc, HitCount desc
```

sigma_soar_blocklist_enterprise_pki_safeguard.yml

SIGMA

```
title: Unfiltered SOAR Threat Feed Auto-Block Missing PKI and Port 3306 Safeguard
id: 7d3a9f12-4c18-4b90-9a31-130000000001
status: stable
description: Detects SOAR or TIP firewall automation jobs pushing raw C2 indicators onto perimeter blocklists
without excluding TCP/3306 MySQL collisions, suspected-only cloud ALB rotations, dual-use OAST hunt leads, or
dynamic telecom subscriber ASNs.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.command_and_control
  - attack.t1071.001
logsource:
  category: application
  product: soar_automation
detection:
  selection_autoblock:
    action_type:
      - 'firewall_block_ip'
      - 'edr_network_indicator_block'
  collision_conditions:
    - indicator_port: 3306
      indicator_framework|contains: 'daybreak'
    - indicator_status: 'suspected'
      indicator_asn|contains: 'Amazon'
    - indicator_handling:
        - 'candidate'
        - 'known_collision'
        - 'subscriber_isp_watchlist'
        - 'dual_use_oast_hunt'
  condition: selection_autoblock and 1 of collision_conditions
falsepositives:
  - Emergency manual incident-response containment overrides approved by SOC leadership
level: high
---
title: Outbound Host Connection to Confirmed C2 Endpoint Port Profile
id: 7d3a9f12-4c18-4b90-9a31-130000000002
status: stable
description: Detects non-browser system or scripting processes initiating outbound connections to corroborated,
confirmed C2 infrastructure. Matches on confirmed C2 addresses are never suppressed.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.command_and_control
  - attack.t1071.001
  - attack.t1583.006
logsource:
  category: network_connection
  product: windows
detection:
  selection_Initiated:
    Initiated: 'true'
    Image|endswith:
      - '\powershell.exe'
      - '\pwsh.exe'
      - '\cmd.exe'
      - '\rundll32.exe'
      - '\regsvr32.exe'
      - '\curl.exe'
```

```
      - '\mshta.exe'
  selection_ti_match:
    DestinationIp:
      - '64.227.189.225'
      - '128.199.131.70'
      - '167.71.229.25'
      - '95.156.207.122'
      - '201.79.4.122'
      - '188.68.60.39'
      - '46.105.117.146'
      - '103.137.250.153'
  condition: selection_Initiated and selection_ti_match
falsepositives:
  - Authorized threat-intelligence sandbox verification probes isolated from corporate VLANs
level: critical
---
title: Suspicious Archiving or Transfer Process Access to Live AI Agent Credential Stores
id: 7d3a9f12-4c18-4b90-9a31-130000000003
status: stable
description: Detects suspicious read access to live autonomous coding agent credential stores
(.claude, .cursor, .codex, .openclaw, .hermes/sessions) by archiving, network-transfer, or scripting processes.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.credential_access
  - attack.t1552.001
logsource:
  category: file_access
  product: linux
  definition: Requires Linux auditd, eBPF file-access telemetry, or Sysmon-for-Linux open/read audit events
capturing process Image, AccessMask/Syscall, and TargetFilename.
detection:
  selection_agent_paths:
    TargetFilename|contains:
      - '/.claude/.credentials.json'
      - '/.claude/projects/'
      - '/.cursor/mcp.json'
      - '/.openclaw/identity/device.json'
      - '/.hermes/sessions/'
      - '/.hermes/.env'
  selection_suspicious_archive_or_transfer_process:
    Image|endswith:
      - '/tar'
      - '/zip'
      - '/7z'
      - '/gzip'
      - '/curl'
      - '/wget'
      - '/scp'
      - '/rsync'
      - '/nc'
      - '/ncat'
      - '/socat'
      - '/python3'
      - '/python'
      - '/base64'
  selection_read_access:
    AccessMask|contains:
      - 'READ'
      - 'O_RDONLY'
      - 'O_RDWR'
      - 'open'
      - 'openat'
  condition: selection_agent_paths and selection_suspicious_archive_or_transfer_process and selection_read_access
falsepositives:
  - Authorized workstation backup agents reading developer home directories under approved change control
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/c2-threat-feed-collisions/
Short Read · brandefense.io/odin-research/blog/when-threat-feeds-call-benign-servers-c2/
Defense Kit · brandefense.io/odin-research/defense-kit/