ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 48 / 30 · CREDENTIAL THEFT

Exposed OAuth2, JWT, SAML and Merchant Keys
A Defender Audit

How 49 misconfigured web servers expose OAuth2, JWT, SAML, and merchant signing keys alongside multi-gigabyte debug logs

Two-week observation window ending 21 November 2026 · Prepared November 2026
Full Report · Public edition · Version 1.1

752
Sampled files across 7 signing-key, certificate, SAML cache, and debug-log classes (0 hashes published)

120 / 24
Corroborating private-key and cloud OAuth records sampled across 24 separate servers

541 vs 2
Non-fixture server, reverse-proxy, and CA PEM key entries separated from unit-test signing fixtures

0
Published indicators of compromise

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

How 49 misconfigured web servers expose OAuth2, JWT, SAML, and merchant signing keys alongside multi-gigabyte debug logs

01	Executive Summary	04
02	Fourteen-Day Secret Findings and the Seven-Surface Signing Model	07
03	Laravel Passport OAuth2 Keys, LexikJWT, and Debug-Log Overlap	11
04	Symfony LightSAML Caches, Merchant mTLS Bundles, and PKCS#12 Keystores	15
05	Private-Key Dominance, Multi-Gigabyte Logs, and Owner Boundaries	19
06	What Defenders Should Do	22
07	Detection Rules	24
08	Indicators	27
09	Scope and Limits	30
10	Sources	32
A	Indicators of compromise (0)	34
B	Detection rules (4 files)	35

IN SHORT

In a two-week observation window ending 21 November 2026, Odin by Brandefense observed 49 misconfigured open web directories reporting 103,410 files and exposing 752 signing-key, certificate, SAML cache, and debug-log files across seven classes. All 49 servers belong to misconfigured users and organizations with zero adversary attributions, so all IP addresses and file hashes are withheld. Together they expose Laravel Passport keys, Symfony LexikJWT keys, LightSAML caches, merchant mTLS bundles, and debug logs. Lock web document roots to public folders and move signing keys into KMS vaults.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 21 November 2026, Odin by Brandefense observed 49 misconfigured open web directories reporting 103,410 files and exposing 752 signing-key, certificate, SAML cache, and debug-log files across seven classes. All 49 servers belong to misconfigured users and organizations with zero adversary attributions, so all IP addresses and file hashes are withheld. Together they expose Laravel Passport keys, Symfony LexikJWT keys, LightSAML caches, merchant mTLS bundles, and debug logs. Lock web document roots to public folders and move signing keys into KMS vaults.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Single sign-on federation, stateless REST APIs, and payment settlement gateways rely on asymmetric private keys to vouch for user identities and financial payouts. Across a two-week observation window ending 21 November 2026, Odin by Brandefense reviewed 49 misconfigured application and developer web servers reporting 103,410 files. Those servers exposed 752 signing-key, keystore, SAML cache, and debug-log artifacts. Signing keys exposed this way belong on any dark web monitoring watch list.

Across our broader 14-day set of 800 secret findings spanning 28 secret types, asymmetric private keys ranked first at 39.50% (316 findings). We opened zero key bodies, forged zero tokens, treated all hosts as misconfigured owners rather than operators, and withheld all addresses and hashes.

1. **Asymmetric private keys lead 14-day secret leaks while standard web ports drive exposure.**
   Private keys account for 316 of 800 14-day secret findings (39.50%), outnumbering database connection strings (121) and AWS access keys (113) combined (234). Among the 49 open-directory signing-key servers, 42 hosts (85.71%) listen on standard HTTP or HTTPS ports 80 and 443 rather than obscure debug ports.

2. **Every exposed Laravel Passport signing-key server also exposes application debug logs.**
   Six Apache servers expose 15 Laravel Passport OAuth2 key files, including nine RSA private keys sized 3,292 to 3,322 bytes. Separate hosts expose Symfony LexikJWT keys and compiled LightSAML container caches. All six Passport servers (100%) also expose framework debug log files reaching 2.47 GB on Host A. Whether those logs contain token claims was not verified.

3. **Merchant mTLS bundles, PKCS#12 keystores, and test fixtures require context-aware triage.**
   Public upload and certificate directories expose payment-gateway client bundles (three files on Host D in China) alongside seven `.pfx` keystores and 543 PEM keys. Two merchant keys in a separate 120-record sample corroborate that pattern. Separating two unit-test signing fixtures (Host F) from 541 server and CA keys prevents false breach alerts.

Three Decisions for Leaders

1. **Restrict Laravel, Symfony, and Craft web roots to their dedicated public directories.** Keep application storage, JWT configuration, compiled cache, and TLS key folders outside all HTTP document roots, and disable directory auto-indexing on every virtual host.
2. **Move OAuth2, JWT, SAML, and merchant mTLS signing keys into KMS or HSM vaults.** Replace flat `.key`, `.pem`, `.p12`, and `.pfx` files on disk with cloud KMS signing APIs, publish `kid`-pinned JSON Web Key Sets, and restrict merchant mTLS calls by source IP.
3. **Turn off production debug logs and filter unit-test key fixtures during triage.** Set `APP_DEBUG=false` for production error display, and configure application loggers to redact bearer tokens and claims, while separating test-fixture sample keys from non-fixture key incidents.

The Study in Numbers

Number	What it is
49 / 103,410	Misconfigured open-directory servers and combined reported files reviewed (0 adversary hosts)
752	Sampled files across 7 signing-key, certificate, SAML cache, and debug-log classes (0 hashes published)
316 / 800 (39.50%)	Share of 14-day secret findings held by `private_key` (rank 1 across 28 secret types)
120 / 24	Corroborating private-key and cloud OAuth records sampled across 24 separate servers
6 of 6 (100%)	Laravel Passport signing-key servers that co-expose framework debug log files
18.35 GB / 4.40 GB	Total sampled debug-log volume across 177 files and single largest exposed debug log
541 vs 2	Non-fixture server, reverse-proxy, and CA PEM key entries separated from unit-test signing fixtures

How to use this report.

Security leaders and identity architects can start with Chapters 1, 5, and 6. Application, IAM, and payment engineers should read Chapters 3, 4, and 6. SOC analysts and detection engineers can use cyber threat intelligence to prioritize alerts, alongside Chapters 7 and 8 and Appendix B. Scope boundaries and sources sit in Chapters 9 and 10.

02

CHAPTER 02

Fourteen-Day Secret Findings and the Seven-Surface Signing Model

IN SHORT

In the two-week window ending 21 November 2026, `private_key` ranked first across 28 secret types at 39.50% (316 of 800 findings), backed by 120 sampled crypto and OAuth rows across 24 servers. Reviewing open web directories isolated 49 misconfigured servers reporting 103,410 files and 752 artifacts across seven classes. Audit standard HTTP and HTTPS virtual hosts for exposed application storage, JWT configuration, compiled cache, and TLS key folders.

INSIDE THIS CHAPTER

The broader 14-day secret-finding context 07

Forty-nine misconfigured signing-key and debug-log servers 08

02 Fourteen-Day Secret Findings and the Seven-Surface Signing Model

Stateless bearer tokens and mutual TLS (`mTLS`) let microservices, single sign-on portals, and payment APIs verify requests without querying a central session store on every call (T1552.004). That design places immense trust in small private-key files on disk. Unlike a leaked password affecting one user, an exposed OAuth2 signing key, JWT key, or SAML credential store weakens authentication trust across an entire application (T1606.002).

When an Apache, Nginx, or Python web root exposes its parent project tree or certificate folder without authentication, those signing artifacts become readable from the internet. We reviewed file paths, byte sizes, and secret-type counts statically without opening key bodies, testing passphrases, forging tokens, or contacting any server.

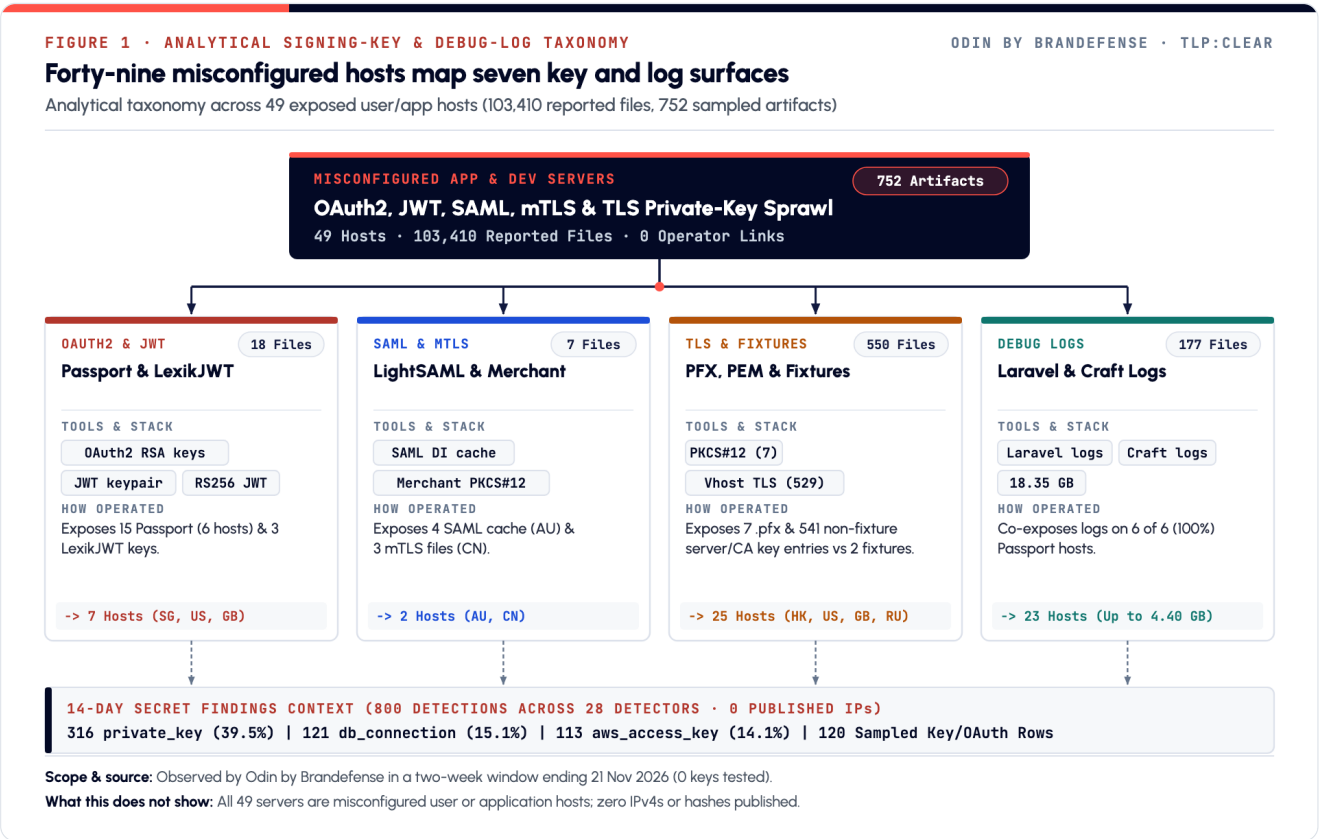


Figure 1: Forty-nine misconfigured hosts map seven key and log surfaces

The broader 14-day secret-finding context

During the two-week observation window ending 21 November 2026, Odin by Brandefense recorded 800 secret findings across 28 secret types. Asymmetric `private_key` findings ranked first overall with 316 detections (39.50%), exceeding database URLs (121, 15.13%) and AWS access keys (113, 14.13%) combined (234, 29.25%). Bot, wallet, exchange, and cloud OAuth secrets follow, led by `telegram_bot_token` (59), `bip39_mnemonic` (36), `github_token` (22), `binance_key` (17), and `th_coinbase_1` (12). Next come `th_azuresastoken` (11), `gcp_oauth_secret` (10), `square_token` (10), `th_azure_openai` (9), and `gcp_service_account` (8).

We also reviewed a 120-record sample across 24 additional servers (0 overlap with the 49 directory servers; addresses withheld). It holds 116 `private_key` rows (96.67%), 2 `gcp_service_account` rows (1.67%), and 2 `gcp_oauth_secret` rows (1.67%). Across 10 geographies, China (62), Hong Kong (37), and India (11) lead, followed by 2 each in Germany, Spain, and Great Britain. Vietnam, Indonesia, Italy, and the US account for 1 each.

Secret type (28 total)	14-day count (800)	Share of 14d	120-row sample	Sample servers (24)	Credential or signing role
`private_key`	316	39.50%	116	23	PEM RSA, PKCS#8, EC, TLS, JWT, and merchant mTLS private keys (T1552.004)
`db_connection`	121	15.13%	0	0	Relational, document, and cache database connection URLs (T1552.001)
`aws_access_key` (+ 3 secret)	113	14.13%	0	0	AWS IAM access keys (113) alongside 3 `aws_secret_key` findings
`telegram_bot_token`	59	7.38%	0	0	Automated messaging bot and webhook API tokens (T1528)
`bip39_mnemonic` (+ 8 ETH)	36	4.50%	0	0	Wallet seed phrases (36) alongside 8 `eth_private_key` findings
`github_token` (+ 6 PAT)	22	2.75%	0	0	GitHub tokens (22) plus 6 `github_pat` personal access tokens
`binance_key` (+ 12 Coinbase)	17	2.13%	0	0	Exchange and payment keys (17 Binance, 12 Coinbase, 10 Square, 4 Stripe)
`th_azuresastoken` & Azure keys	11	1.38%	0	0	Azure SAS (11), Azure OpenAI (9), APIM (2), ACR (1), Function (1)
`gcp_oauth_secret`	10	1.25%	2	1	Google Cloud OAuth2 client secrets (1 GB server; port withheld)
`gcp_service_account` (+ 8 ADC)	8	1.00%	2	2	GCP service-account JSON keys (Spain and HK; ports withheld) plus 8 ADC
Remaining 18 secret types	87	10.88%	0	0	Discord (8), Alibaba (7), Slack (2), Tailscale (2), CS (2), CF/DO (2)

Forty-nine misconfigured signing-key and debug-log servers

Reviewing open web directories for OAuth2/JWT keys, SAML caches, merchant mTLS bundles, PKCS#12 keystores, PEM keys, and debug logs identified 49 servers (addresses, ports, paths, and filenames withheld). Together they report 103,410 files (244 to 10,000 per host) and 752 sampled artifacts (18,346,654,294 bytes, or 18.35 GB).

1. **Standard web ports account for 85.71% of hosts.** Twenty-three servers (46.94%) listen on port 443 and 19 (38.78%) listen on port 80 (42 of 49). Seven servers (14.29%) use ports 8000 (2), 8888 (2), 8080 (1), 8081 (1), and 81 (1); host-to-port assignments are withheld.
2. **Apache and Nginx make up 93.88% of hosts.** Apache runs on 28 hosts (57.14%), Nginx on 18 (36.73%), and Python HTTP on 3 (6.12%). Across 10 geographies, the US (17, 34.69%) and Hong Kong (16, 32.65%) lead, followed by Singapore (5), India (3), Great Britain (2), and Canada (2). China, Russia, the UAE, and Australia account for 1 each. Nine hosts (18.37%) expose two classes simultaneously.

Exposure class	Sampled files (752)	Share	Servers (49)	Size range (bytes)	Total bytes	Main misconfiguration pattern
Generic PEM or TLS private keys	543	72.21%	24	32 to 3,212	916,247	529 HK virtual-host TLS keys, 12 app/CA keys, 2 unit-test fixtures (ports/paths withheld)
Framework debug log sprawl	177	23.54%	23	4,851 to 4,402,341,478	18,345,559,684	171 Laravel debug logs (28 unsized) + 6 Craft CMS debug logs (paths withheld)
Laravel Passport OAuth2 keys	15	1.99%	6	812 to 3,322	34,710	9 OAuth2 RSA private keys + 6 public keys (100% co-exposed with debug logs)
TLS `.pfx` private keystores	7	0.93%	2	3,736 to 6,701	29,498	Binary PKCS#12 keystore bundles on 1 US and 1 GB server (ports/paths withheld)
Symfony SAML credential cache	4	0.53%	1	1,126 to 96,832	100,824	Compiled LightSAML DI container service cache files (port/paths withheld)
Payment merchant mTLS certs	3	0.40%	1	1,509 to 2,774	5,987	Payment PKCS#12 client bundle, PEM private key, and X.509 cert (port/paths withheld)
Symfony LexikJWT signing keys	3	0.40%	1	800 to 3,272	7,344	Two RSA private-key entries and one public-key entry on 1 US server (port/paths withheld)

CHAPTER 03

Laravel Passport OAuth2 Keys, LexikJWT, and Debug-Log Overlap

IN SHORT

Pointing a web server document root at a PHP repository folder instead of its public web directory exposes OAuth2 and JWT signing keys directly over HTTP. Across six Laravel Passport servers (15 key files, including 9 RSA private keys sized 3,292 to 3,322 bytes), 100% (6 of 6) also expose framework debug logs reaching 2.47 GB on Host A. Restrict document roots to public directories, move signing keys into KMS, and turn off `APP_DEBUG`.

INSIDE THIS CHAPTER

Laravel Passport OAuth2 signing-key sprawl across six servers 11

Why 100% (6 of 6) debug-log overlap raises offline `RS256` forgery risk 12

Symfony `LexikJWTAuthenticationBundle` keypair on Host B 12

03 Laravel Passport OAuth2 Keys, LexikJWT, and Debug-Log Overlap

PHP web frameworks generate RSA or ECDSA keypairs on local disk to issue stateless JSON Web Tokens (RFC 7519, RFC 6749). Laravel Passport and Symfony `LexikJWTAuthenticationBundle` write 4096-bit RSA private and public key files into application storage or configuration directories (T1552.004).

Both frameworks assume that only the public front-controller directory sits inside the web server's document root, which defenders can audit with external attack surface management. When an administrator points a web server at the parent project directory and leaves directory indexing active, internal storage and JWT configuration folders become readable over HTTP.

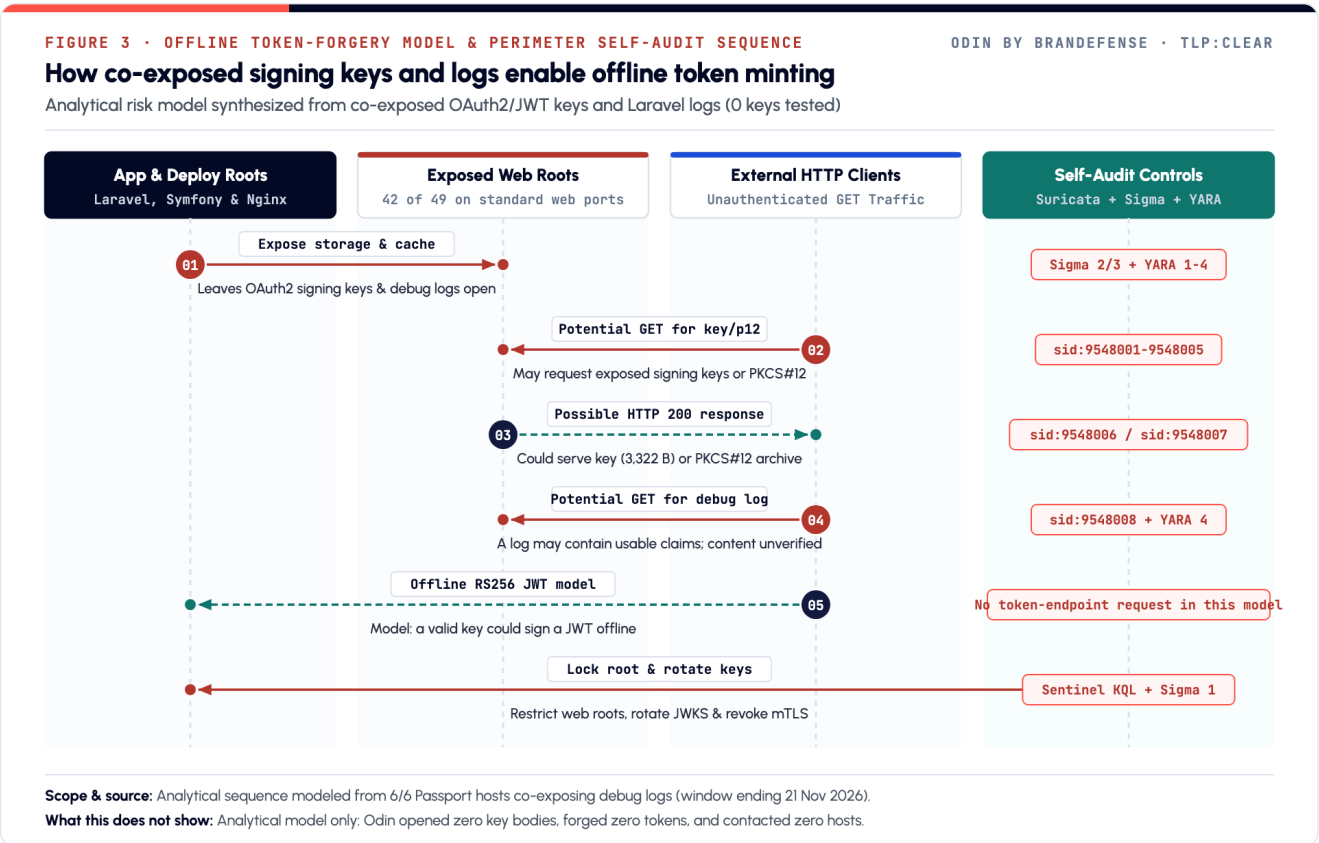


Figure 3: How co-exposed signing keys and logs enable offline token minting

Laravel Passport OAuth2 signing-key sprawl across six servers

Six misconfigured owner servers expose 15 Laravel Passport key files (34,710 bytes total) across Singapore (3), the United States (2), and Great Britain (1). These comprise 9 OAuth2 RSA private-key files and 6 public-key files. Individual server ports, paths, and filenames are withheld.

Byte sizes across all 15 files match standard 4096-bit RSA PEM keys:

- **Private signing keys (9 files across 9 application folders).** Seven OAuth2 private-key files measure exactly 3,322 bytes, while two private-key files on a server in Singapore measure 3,292 bytes.
- **Public verification keys (6 files).** All six OAuth2 public-key files measure 812 bytes.

- **Multi-tenant shared web roots.** Two servers in Singapore host multiple Laravel applications beneath a single open document root. One Singapore server (7,132 reported files) exposed three separate 3,322-byte OAuth2 private keys across three app folders. A second Singapore server (10,000 reported files) exposed two complete private-and-public keypairs. On the Great Britain server (3,075 reported files), an OAuth2 private key sat in the top-level storage tree.

Why 100% (6 of 6) debug-log overlap raises offline `RS256` forgery risk

In an analytical risk model (Odin opened zero files and forged zero tokens), using an exposed OAuth2 private key still requires valid JWT claims (`iss`, `aud`, `sub`, `jti`, `scopes`) (RFC 7519, T1552.004). Because Laravel stores both OAuth2 keys and application logs under the same storage tree, **all six (100%) of the Passport servers co-expose framework debug logs (30 logs)**:

1. **Host A (United States, 10,000 reported files; address, port, paths, and filenames withheld).** This misconfigured owner server exposed an OAuth2 private key (3,322 bytes) and public key (812 bytes) alongside 19 Laravel logs totaling 3,060,213,581 bytes (3.06 GB). Those include a 2,469,606,195-byte (2.47 GB) primary log, a 583.01 MB log, a same-app 1.05 MB log, and 14 daily rotated log files.
2. **Singapore, Great Britain, and second US Passport servers.** One Singapore server (1,541 reported files) paired an OAuth2 private key with a 620,176,544-byte (620.18 MB) debug log in the same app folder. The dual-app Singapore server paired two 3,292-byte keys with 108.13 MB and 5.04 MB logs. The remaining Singapore, Great Britain, and United States servers (502 reported files) also exposed debug logs; all ports, paths, and filenames are withheld.

If an exposed key is valid and a log contains usable claims, an attacker could attempt offline `RS256` signing without querying the token endpoint. Log contents, key validity, and API revocation behavior were not tested.

Symfony `LexikJWTAuthenticationBundle` keypair on Host B

Host B, a misconfigured owner server in the United States, exposed three Symfony JWT key entries totaling 7,344 bytes. Its port, paths, and filenames are withheld.

Symfony `LexikJWTAuthenticationBundle` uses an RSA keypair by default for stateless API JWT authentication (`RS256`, `RS384`, or `RS512`). Across 1,192 reported files on **Host B**, two 3,272-byte private-key entries and one 800-byte public-key entry are consistent with a 4096-bit RSA keypair. Whether the key can sign accepted JWTs, or grant administrative access, was not tested.

Server profile	Host details	Country	Reported files	Key files (bytes)	Co-exposed debug logs	Co-exposure pattern
Host A (Passport + logs)	Port/paths withheld	US	10,000	1 priv (3,322), 1 pub (812)	19 logs (3.06 GB; max 2.47 GB)	Same-app OAuth2 keypair + 1.05 MB & 2.47 GB debug logs
Dual-app Passport server	Port/paths withheld	SG	10,000	2 priv (3,292), 2 pub (812)	2 logs (108.13 MB & 5.04 MB)	Two Passport keypairs paired 1:1 with framework debug logs
Multi-app Passport server	Port/paths withheld	SG	7,132	3 priv (3,322 each)	6 logs (2.15 MB max, 4 unsized)	Three multi-tenant OAuth2 private keys across app trees
Root Passport server	Port/paths withheld	GB	3,075	1 priv (3,322), 1 pub (812)	1 log (unsized)	Top-level OAuth2 keypair exposed from web root
Single-app Passport server	Port/paths withheld	SG	1,541	1 priv (3,322), 1 pub (812)	1 log (620.18 MB)	Same-app OAuth2 keypair + 620.18 MB debug log
Compact Passport server	Port/paths withheld	US	502	1 priv (3,322), 1 pub (812)	1 log (unsized)	Exposed application OAuth2 keypair + debug log
Host B (Symfony LexikJWT)	Port/paths withheld	US	1,192	2 priv (3,272), 1 pub (800)	0 sampled logs	Exposed Symfony JWT private-and-public keypair

CHAPTER 04

Symfony LightSAML Caches, Merchant mTLS Bundles, and PKCS#12 Keystores

IN SHORT

Private-key sprawl also hits enterprise SAML 2.0 federation caches, payment-gateway mutual TLS client bundles, and PKCS#12 archives. Our 49-server cohort includes Symfony `LightSAML` caches on Host C, merchant mTLS bundles on Host D, 7 `.pfx` keystores on Host E, and 541 server PEM keys alongside 2 unit-test fixtures (Host F). Block compiled cache directories, public keystore uploads, and private-key extensions at reverse proxies.

INSIDE THIS CHAPTER

Compiled Symfony `LightSAML` container cache on Host C 15

Payment-gateway merchant mTLS bundles on Host D 16

PKCS#12 `.pfx` keystores (Host E) and 543 PEM keys vs. unit-test fixtures (Host F) 17

04 Symfony LightSAML Caches, Merchant mTLS Bundles, and PKCS#12 Keystores

Beyond OAuth2 and API JWT files, single sign-on and payment settlement systems rely on other asymmetric credentials. These include SAML 2.0 X.509 certificates, payment merchant mTLS client bundles, and RFC 7292 PKCS#12 keystores (T1552.004, T1606.002).



Figure 2: Seven signing-key and debug-log classes map to self-audit rules

Compiled Symfony `LightSAML` container cache on Host C

Host C, a misconfigured owner server in Australia (244 reported files; address, port, paths, and filenames withheld), exposed 4 compiled Symfony SAML cache files totaling 100,824 bytes. Inside an exposed production cache directory, three dependency-injection (`DI`) container PHP service factories sat beside a 96,832-byte compiled container file:

- Own-credential store service factory (1,126 bytes; filename withheld)
- Credential-resolver service factory (1,228 bytes; filename withheld)
- Federation credential-store service factory (1,638 bytes; filename withheld)

Symfony applications using `lightsaml/sp-bundle` or `lightsaml/symfony-bridge` compile SAML parameters into PHP factory classes inside the production container cache. The observed entries correspond to LightSAML credential-store and resolver services. The cache bodies were not inspected, so whether they expose key paths, passphrases, or entity IDs remains unverified.

If a static alias or backup virtual host serves `.php` files as plain text, HTTP `GET` can expose compiled SAML container arguments. Even when PHP runs normally, open directory listings reveal the compiled SAML service inventory (T1606.002). If an application also exposes its SAML signing key, an attacker could attempt to forge `AuthnRequest`, `LogoutRequest`, or `<saml:Assertion>` XML signatures.

Payment-gateway merchant mTLS bundles on Host D

Host D, a misconfigured owner server in China (10,000 reported files; address, port, paths, and filenames withheld), exposed 3 payment-gateway merchant mTLS files totaling 5,987 bytes inside a public upload directory:

- Binary PKCS#12 merchant client certificate and key archive (2,774 bytes; filename and UUID prefix withheld)
- PEM merchant client private key (1,704 bytes; filename withheld; key type, bit length, and encryption status unverified)
- PEM-encoded X.509 merchant client certificate (1,509 bytes; filename withheld)

Asian mobile-payment and merchant-acquiring SDKs issue this three-file bundle to authenticate mutual TLS order refunds, enterprise B2C payouts, and settlement queries (T1552.004). Although `.p12` archives typically carry password wrapping, saving the 1,704-byte PEM private key inside a public upload folder exposes the key file directly; its encryption status was not verified. Our 120-record secret sample independently corroborates this upload pattern on a separate server in China (address, port, and paths withheld) exposing two merchant PEM private-key files across two tenant folders.



Figure 4: Web roots expose Passport, LexikJWT, LightSAML, mTLS, and PFX keys

PKCS#12 `.pfx` keystores (Host E) and 543 PEM keys vs. unit-test fixtures (Host F)

Two remaining classes account for 550 of the 752 sampled files (73.14%):

1. **Binary PKCS#12 `.pfx` keystores (Host E: 7 files across 2 servers, 29,498 bytes).** A misconfigured server in the United States (6,203 reported files) exposed a 6,701-byte `.pfx` archive beside three PEM `.key` files (1,743, 251, and 251 bytes). A server in Great Britain (1,878 reported files) exposed six `.pfx` keystores (3,736 to 3,849 bytes) across a certificate directory and its web root (RFC 7292, T1552.004). All addresses, ports, paths, and filenames are withheld.
2. **Generic PEM and TLS private keys (543 files across 24 servers, 916,247 bytes).** Fifteen servers in Hong Kong (515 to 694 reported files each) exposed 529 TLS key files (522 virtual-host `.key` files and 7 server TLS `.key` files, led by 377 1,705-byte RSA keys). Shared TLS folder structure across those 15 servers and 3 Hong Kong sample servers fits a multi-vhost panel template. Seven standalone servers across the United States, Hong Kong, and Russia exposed 12 keys (32 to 3,212 bytes). Those include **three 227-byte ECDSA P-256 keys (one in a CA subfolder) on a US server** and a 255-byte configuration key on another US server. By contrast, **Host F** comprises two servers in the United States (1,539 reported files each) exposing 1,679-byte unit-test signing-key fixtures. All addresses, ports, paths, and filenames are withheld.

Exposure group	Host details	Country	Files	Size (bytes)	Artifact pattern and posture risk
Host C: LightSAML DI cache	Port/paths withheld	AU	4	1,126 to 96,832	Three compiled LightSAML credential-service classes + container cache
Host D: Merchant mTLS bundle	Port/paths withheld	CN	3	1,509 to 2,774	Public upload folder with PKCS#12 bundle, PEM key (1,704 B), and X.509 cert
Host E: US `.pfx` keystore	Port/paths withheld	US	1	6,701	`.pfx` archive co-exposed beside 3 PEM `.key` files (251 to 1,743 B)
Host E: GB `.pfx` keystores	Port/paths withheld	GB	6	3,736 to 3,849	One certificate-folder `.pfx` file (3,849 B) plus 5 web-root `.pfx` bundles
Clustered vhost TLS keys	Ports/paths withheld (15)	HK	529	1,679 to 1,730	522 virtual-host TLS `.key` files + 7 server TLS `.key` files (35 to 38/host)
Standalone app & CA keys	Ports/paths withheld (7)	US, HK, RU	12	32 to 3,212	Includes CA-scoped ECDSA key (227 B) and configuration key (255 B)
Host F: Unit-test fixtures	Ports/paths withheld (2)	US	2	1,679	Unit-test signing-key fixtures (route to cleanup, not breach paging)

05

CHAPTER 05

Private-Key Dominance, Multi-Gigabyte Logs, and Owner Boundaries

IN SHORT

Pairing our 49-server open-directory cohort (752 files) with the 800-finding 14-day secret dataset and 120-row sample shows two clear takeaways. Asymmetric private keys form the single largest secret-leak class (39.50%), while naive `.pem` and `.key` checks mix unit-test fixtures with live application OAuth2 and JWT signing keys. Pair context-aware key triage with strict log rotation.

INSIDE THIS CHAPTER

Multi-gigabyte Laravel and Craft debug-log sprawl (177 files, 18.35 GB) 19

Separating unit-test fixtures from non-fixture key entries and corroborating the 120-row sample 20

Owner-versus-operator boundaries 20

05 Private-Key Dominance, Multi-Gigabyte Logs, and Owner Boundaries

Security teams often focus secret-scanning budgets on cloud IAM keys and database URLs while treating `.pem` and `.key` alerts on web servers as low-priority TLS noise. In our 14-day dataset, `private_key` findings (316 of 800, 39.50%) outnumber `db_connection` (121) by 2.61x and `aws_access_key` (113) by 2.80x.

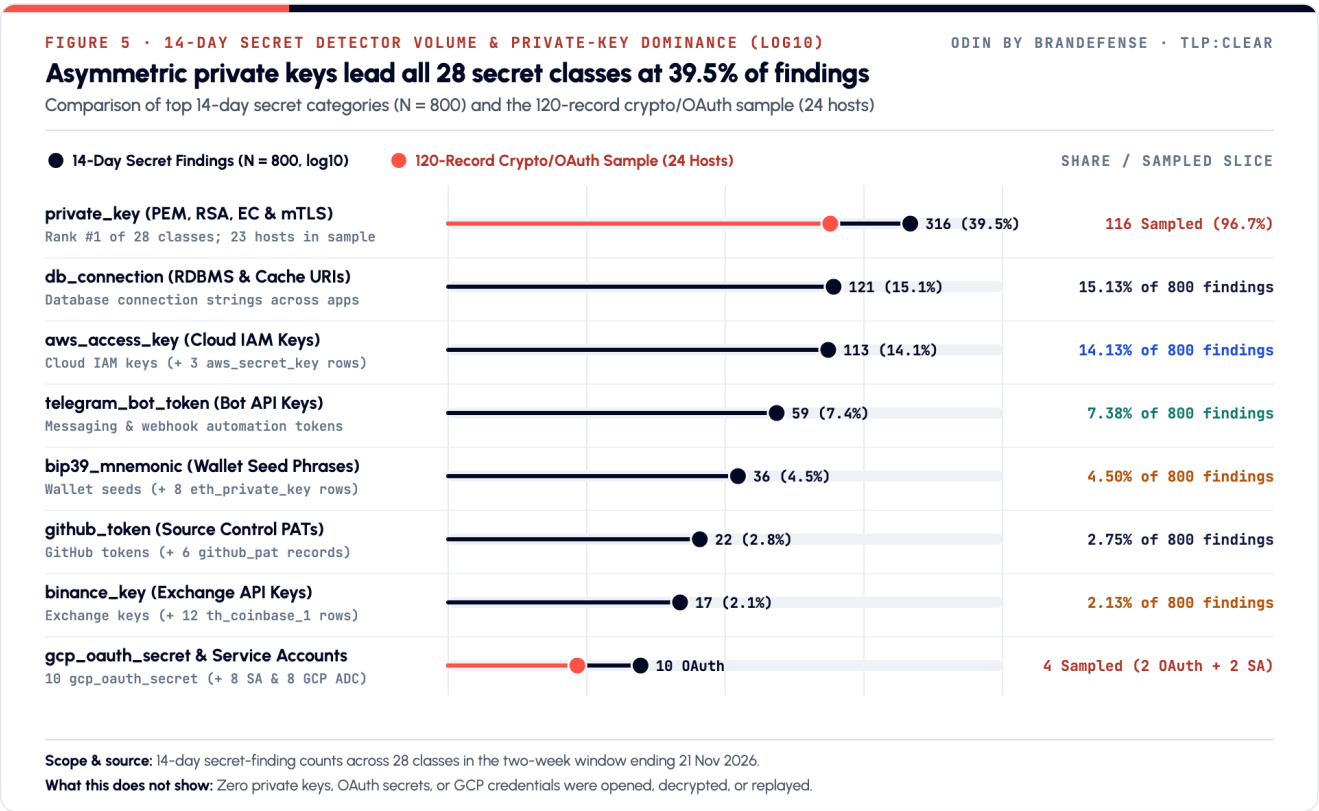


Figure 5: Asymmetric private keys lead all 28 secret classes at 39.5% of findings

Multi-gigabyte Laravel and Craft debug-log sprawl (177 files, 18.35 GB)

Across the 49 open-directory servers, 23 hosts (46.94%: 22 Apache and 1 Nginx across the US, Singapore, India, Canada, Great Britain, and the UAE) expose 177 framework debug logs. Those 177 entries (149 sized from 4,851 to 4,402,341,478 bytes plus 28 unsized entries) total 18,345,559,684 bytes (18.35 GB decimal, or 99.994% of all sampled bytes):

- **Laravel debug logs (171 files across 22 servers).** Five Laravel log files exceed 1.0 GB, and 15 exceed 100 MB. A server in Singapore holds the largest file at 4,402,341,478 bytes (4.40 GB). Next come 3.48 GB on a US server, 2.47 GB on **Host A**, and 2.04 GB plus 1.72 GB on another US server. That US server also exposed a 104.86 MB daily log inside its public storage tree after an administrator linked the full internal storage folder into the public directory. All ports, paths, and filenames are withheld.
- **Craft CMS / Yii2 web debug logs (6 files on 1 US server).** Co-exposed beside a 255-byte configuration `.key` file sat six daily web debug log files (19.15 MB to 112.79 MB, 454.41 MB total). All ports, paths, and filenames are withheld, and the log bodies were not inspected.

Separating unit-test fixtures from non-fixture key entries and corroborating the 120-row sample

Deploying third-party dependency or test directories inside a web root exposes bundled RSA unit-test keys. Examples include open-source library test keys and the two 1,679-byte unit-test signing-key fixtures on **Host F** in the United States (ports, paths, and filenames withheld). Production OAuth2, JWT, SAML, and mTLS endpoints do not trust unit-test fixtures unless fatally misconfigured.

Route test-fixture matches to web-root cleanup rather than emergency breach paging. Reserve urgent rotation for the 541 non-fixture server and CA PEM key entries, 9 OAuth2 private keys, and 2 JWT private-key rows. Apply the same priority to the 4 LightSAML caches, 3 merchant mTLS files, and 7 `.pfx` keystores.

The 120-record sample across 24 separate servers reinforces the same pattern across ports 8000 (49), 443 (38), 80 (23), 10081 (4), 8080 (3), 554 (2), and 9999 (1); host-level ports and paths are withheld.

Server group in 120-row sample	Servers (24)	Records (120)	Secret types matched	Exposed artifact classes (ports/paths withheld)
China multi-key server	1	49	`private_key` (49)	Certificate-directory `.key` and `.pem` private-key trees
Hong Kong virtual-host TLS cluster	3	35	`private_key` (35)	Reverse-proxy and virtual-host TLS private-key files
China application and merchant servers	4	13	`private_key` (13)	Two multi-tenant merchant mTLS private keys + 11 PEM keys
India application web servers	7	11	`private_key` (11)	App-scoped `.key`, `.pem`, and configuration key files
GCP OAuth & service-account hosts	3	6	2 OAuth, 2 SA, 2 key	GB (2 OAuth), Spain (1 SA, 1 key), HK (1 SA, 1 key)
DE, US, VN, ID, IT web servers	6	6	`private_key` (6)	Data-directory and application PEM private-key files

Owner-versus-operator boundaries

1. **All 73 servers (49 directory hosts plus 24 sample hosts) are misconfigured owners.** Each reflects an unlocked web root, mislinked storage directory, exposed compiled container cache, public upload folder, or ad-hoc HTTP listener. That includes the 15 Hong Kong virtual-host TLS servers (0 offensive tools observed; all ports, paths, and filenames withheld).
2. **Zero keys opened or tokens forged.** Odin tested 0 passphrases, decrypted 0 `.p12` / `.pfx` bundles, and forged 0 OAuth2, JWT, SAML, or mTLS requests.

CHAPTER 06

What Defenders Should Do

IN SHORT

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

06 What Defenders Should Do

Web platform and virtual-host engineering

- [ ] Point Laravel, Symfony, and Craft web roots strictly at their dedicated public web directories and disable directory indexing.
- [ ] Never symlink internal application storage trees into public web folders, and block storage, JWT configuration, compiled cache, TLS key, and certificate directories at reverse proxies.
- [ ] Shut down ad-hoc Python HTTP file listeners on non-standard ports, and strip dependency and unit-test fixture folders from production web builds.

Identity, OAuth2, JWT, and SAML architecture

- [ ] Move Laravel Passport OAuth2, Symfony LexikJWT, and LightSAML signing keys out of flat files into non-exportable cloud KMS or HSM signing APIs.
- [ ] Publish `kid`-pinned JWKS endpoints so signing keys can rotate cleanly, and enforce `jti` token revocation checks on sensitive API endpoints.

Payment security, logging, and SOC

- [ ] Never save payment-gateway merchant mTLS bundles inside public upload trees. Move merchant keys into secret vaults, enforce payment API caller IP allowlists, and revoke any exposed client certificate.
- [ ] Set `APP_DEBUG=false` in production and redact `Authorization` and `Cookie` headers in application loggers. Enforce size-capped log rotation so multi-gigabyte framework debug logs never build up, and deploy the rules in Chapter 7.

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 24

How the KQL, Sigma, and YARA rules work 25

MITRE ATT&CK mapping 25

07 Detection Rules

The full rules are in Appendix B. Static syntax checks (`lint_suricata.py` with 0 problems, plus modifier and fixture checks) passed cleanly. Suricata, YARA, Sigma, and Kusto engines were not installed in this environment, and rules were not replayed against live traffic. Test and tune them in staging before production inline blocking.

Rule file	What it catches
`odin_exposed_oauth_jwt_saml_signing_keys.rules`	Suricata 7: eight rules for inbound requests on `.key`, `.pem`, `.p12`, `.pfx`, and `.log` URIs plus outbound HTTP 200 PEM, compiled SAML cache, and PKCS#12 responses
`sentinel_exposed_signing_keys_and_saml_cache_hunt.kql`	Sentinel and Defender XDR: `http.server`, key-generation commands, or `.key`, `.pem`, `.p12`, `.pfx`, and `.log` processes joined with active web ports
`sigma_exposed_oauth_jwt_saml_and_merchant_keys.yml`	Sigma (3 rules): web HTTP 200/206 key, keystore, or log downloads, Linux `http.server` or key `chmod` commands, and key files written inside web document roots
`odin_exposed_signing_keys_saml_and_merchant_certs.yar`	YARA (4 rules): PEM RSA/EC private keys, compiled Symfony LightSAML cache classes, RFC 7292 PKCS#12 keystores, and Laravel/Craft auth debug logs under 50 MB; use URI or webserver checks for larger logs

How the Suricata signatures work

SID	Looks for	Why
9548001	Inbound `GET` ending in `.key`	Catches HTTP requests for private-key file extensions (block candidate)
9548002	Inbound `GET` ending in `.pem`	Catches HTTP requests for PEM key or certificate extensions (block candidate)
9548003	Outbound 200 body with `Symfony\Component\DependencyInjection` and `LightSaml\Credential\`	Self-audit alert when an internal host serves compiled LightSAML container cache classes
9548004	Inbound `GET` ending in `.p12`	Catches HTTP requests for PKCS#12 `.p12` client bundles (block candidate)
9548005	Inbound `GET` ending in `.pfx`	Catches HTTP requests for binary PKCS#12 `.pfx` keystores (block candidate)
9548006	Outbound 200 body containing `-----BEGIN ... PRIVATE KEY-----`	Self-audit alert when an internal host serves a PEM private key
9548007	Outbound 200 body with ASN.1 30 82, 02 01 03 30, and `pkcs7-data` OID	Self-audit alert when an internal host serves a PKCS#12 archive
9548008	Inbound `GET` ending in `.log`	Catches HTTP requests for application debug-log file extensions (block candidate)

Deploy SIDs 9548001, 9548002, 9548004, 9548005, and 9548008 as alerts; add separately validated proxy or WAF blocks for those URI extensions. Treat any SID 9548003, 9548006, or 9548007 outbound 200 alert on your network as a self-audit exposure check, verifying first whether a matched key sits inside a unit-test fixture folder.

How the KQL, Sigma, and YARA rules work

The Sentinel KQL query looks back 7 days across `DeviceProcessEvents` and `DeviceNetworkEvents`. It finds `python`, `php`, `openssl`, `chmod`, `cp`, `mv`, `tar`, `nginx`, or `apache2` commands referencing `http.server`, `passport:keys`, `lexik:jwt:generate-keypair`, `pkcs12 -export`, `genrsa`, `ecparam`, or `.key`, `.pem`, `.p12`, `.pfx`, and `.log` extensions. It joins those processes with active sockets on ports 80, 443, 8000, 8080, and 8443 outside two build hosts (`pki-vault-builder-01.corp[.]example` and `ci-artifact-signer-01.corp[.]example`).

Sigma rule 1 flags web HTTP 200/206 responses on URIs ending in `.key`, `.pem`, `.p12`, `.pfx`, or `.log`. Sigma rule 2 flags Linux `http.server` or key `chmod` commands outside the two build hosts, and Sigma rule 3 flags `.key`, `.pem`, `.p12`, or `.pfx` files written under `/var/www/`, `/public_html/`, or `/htdocs/`.

YARA rule	Fires when
`ODIN_Exposed_PEM_RSA_Or_EC_Private_Signing_Key`	Under 64 KB, PEM `BEGIN` / `END` private key blocks pair with ASN.1 Base64 markers
`ODIN_Compiled_Symfony_LightSAML_Credential_Store_Cache`	Under 256 KB, `LightSaml\Credential\`, `X509Credential`, or `KeyHelper::createPrivateKey` pair with Symfony DI container markers
`ODIN_PKCS12_Merchant_Or_TLS_Keystore_Bundle`	Under 128 KB, RFC 7292 PKCS#12 v3 bytes pair with `pkcs7-data` and bag OIDs
`ODIN_Exposed_Laravel_Or_Craft_Debug_Log_With_Auth_Context`	Under 50 MB, Laravel or Craft error markers pair with `Authorization`, `Bearer eyJ`, or `League\OAuth2\Server`

MITRE ATT&CK mapping

Tactic	Technique	Observed exposure pattern	Detection
Credential Access	T1552.004 Private Keys	9 OAuth2 private keys, 2 JWT private keys, 3 merchant mTLS files, 7 `.pfx`, 541 PEM keys	SIDs 9548001, 9548002, 9548004 to 9548007; Sigma 1 to 3; YARA 1 & 3; KQL
Credential Access	T1606.002 SAML Tokens (risk model)	Compiled LightSAML DI container cache; SAML key exposure and token forgery unverified	SID 9548003; YARA 2 (compiled container content checks)
Credential Access	T1552.001 Credentials In Files	177 Laravel & Craft debug logs (18.35 GB, 6 of 6 Passport hosts)	SID 9548008; Sigma 1; KQL; YARA 4 only for files under 50 MB

CHAPTER 08

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Structural self-audit patterns in this report 27

08 Indicators

Appendix A holds 0 published indicators: 0 network addresses and 0 file hashes. All 49 open-directory server addresses, all 24 sample server addresses, and all file SHA-256 digests, ports, paths, and filenames are withheld because every host is a misconfigured owner rather than adversary infrastructure. Eighteen structural artifact and URI classes appear in this report only for defensive self-audit.

Type	Count	Where to use it
Network addresses (Appendix A)	0	All 73 misconfigured server addresses are withheld
File hashes (Appendix A)	0	All private-key, certificate, cache, and log hashes are withheld
Self-audit artifact and URI classes (in text)	17	Audit owned web roots, compiled caches, and public upload folders
Unit-test fixture class (in text)	1	Separate unit-test fixture sample keys from non-fixture key alerts

Structural self-audit patterns in this report

Pattern	Type	Exposure surface	Defensive self-audit use
OAuth2 private-key exposure	URI class	Laravel Passport OAuth2	Audit owned web roots for accessible signing keys; rotate exposed keys
OAuth2 public-key exposure	URI class	Laravel Passport OAuth2	Audit owned web roots for exposed OAuth2 verification keys and storage trees
API JWT private-key exposure	URI class	Symfony LexikJWT	Audit owned web roots for accessible JWT signing keys; rotate keypairs
API JWT public-key exposure	URI class	Symfony LexikJWT	Remove JWT configuration directories from web document roots
Compiled DI cache directory exposure	Directory class	Symfony LightSAML cache	Block HTTP access to compiled production container cache directories
SAML own-credential store cache	Artifact class	Symfony LightSAML cache	Audit web roots for compiled LightSAML own-credential store classes
SAML federation credential-store cache	Artifact class	Symfony LightSAML cache	Audit web roots for compiled LightSAML federation credential stores
SAML credential-resolver cache	Artifact class	Symfony LightSAML cache	Audit web roots for compiled LightSAML credential-resolver classes
Merchant mTLS PKCS#12 bundle	Artifact class	Payment merchant mTLS	Block public downloads; revoke and reissue merchant payment mTLS bundles
Merchant mTLS PEM private key	Artifact class	Payment merchant mTLS	Purge exposed merchant PEM private-key files from public upload trees
Merchant mTLS X.509 certificate	Artifact class	Payment merchant mTLS	Audit public upload folders for merchant X.509 client certificates
TLS PKCS#12 keystore directory	Directory class	TLS `.pfx` keystores	Block certificate directories and web-root `.pfx` keystore downloads
Reverse-proxy TLS private key	URI class	Reverse-proxy PEM keys	Move virtual-host and web-server TLS keys outside web document roots

Pattern	Type	Exposure surface	Defensive self-audit use
Data-directory PEM private key	URI class	Generic PEM private keys	Scan application data and CA subfolders for exposed PEM private keys
Primary Laravel debug-log exposure	URI class	Framework debug logs	Block public log access; turn off `APP_DEBUG` and rotate logs
Daily rotated Laravel debug logs	URI class	Framework debug logs	Block HTTP access to dated application debug-log archives
Daily rotated Craft/Yii2 debug logs	URI class	Framework debug logs	Block HTTP access to Craft CMS and Yii2 web debug-log dumps
Unit-test signing-key fixture	Fixture class	Unit-test calibration	Separate unit-test sample keys from non-fixture signing-key alerts

Withheld: all 49 open-directory server IPv4s, all 24 sample server IPv4s, host ports, file paths, filenames, organization folder names, merchant UUID prefixes, compiled container hashes, secret values, and file SHA-256 digests.

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.** All observations come from a two-week observation window ending 21 November 2026.
- **Static review, zero keys opened or tokens forged.** File paths, sizes, and secret counts were reviewed statically without opening private keys, testing passphrases, forging tokens, or contacting any server. Passphrase strength and API `jti` checks are unmeasured.
- **Offline forgery sequence is an analytical model.** Figure 3 models how co-exposed OAuth2 private keys and framework debug logs interact architecturally; Odin performed zero token minting.
- **Misconfigured owners, not operators.** All 49 open-directory hosts and 24 sample hosts are treated as user, developer, or organization servers, including the 15 Hong Kong virtual-host TLS servers. All host addresses, host-linked ports, paths, filenames, and file hashes are withheld.
- **Server file dates vs observation window.** Upload directories (on Host D) and daily rotated framework debug logs carry server dates on disk (withheld in this public edition). Earlier 2026 first-seen timestamps on directory records are historical metadata, never Odin monitoring before November 2026.
- **Rule testing.** Static checks (`lint_suricata.py` with 0 problems, plus modifier and fixture checks) passed on 21 November 2026. Suricata, YARA, Sigma, and Kusto engines were not installed locally, so validate rules before production enforcement.

10

CHAPTER 10

Sources

IN SHORT

This chapter lists the public references used in the report.

10 Sources

1. MITRE ATT&CK. T1552.004 Unsecured Credentials: Private Keys. https://attack.mitre.org/techniques/T1552/004/ (verified 21 November 2026)
2. MITRE ATT&CK. T1606.002 Forge Web Credentials: SAML Tokens. https://attack.mitre.org/techniques/T1606/002/ (verified 21 November 2026)
3. MITRE ATT&CK. T1552.001 Unsecured Credentials: Credentials In Files. https://attack.mitre.org/techniques/T1552/001/ (verified 21 November 2026)
4. MITRE ATT&CK. T1528 Steal Application Access Token. https://attack.mitre.org/techniques/T1528/ (verified 21 November 2026)
5. IETF RFC 7519. JSON Web Token (JWT). https://www.rfc-editor.org/rfc/rfc7519 (verified 21 November 2026)
6. IETF RFC 6749. The OAuth 2.0 Authorization Framework. https://www.rfc-editor.org/rfc/rfc6749 (verified 21 November 2026)
7. IETF RFC 7292. PKCS #12: Personal Information Exchange Syntax v1.1. https://www.rfc-editor.org/rfc/rfc7292 (verified 21 November 2026)
8. OASIS Standard. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf (verified 21 November 2026)
9. Laravel Documentation. Laravel Passport OAuth2 Server (v11.x). https://laravel.com/docs/11.x/passport (verified 21 November 2026)
10. Symfony Documentation. LexikJWTAuthenticationBundle. https://symfony.com/bundles/LexikJWTAuthenticationBundle (verified 21 November 2026)
11. OWASP Cheat Sheet Series. OAuth 2.0 Protocol Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/OAuth2_Cheat_Sheet.html (verified 21 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

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

INSIDE

A. Indicators of compromise (0) 34

B. Detection rules (4 files) 35

A Indicators of Compromise

No individual addresses or file hashes are published for this study. The hosts in this report are exposed or unattributed systems rather than confirmed adversary infrastructure, so they stay anonymous. Use the detection rules in Appendix B.

B Detection Rules

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

odin_exposed_oauth_jwt_saml_signing_keys.rules

SURICATA

```
# Odin by Brandefense - Monograph 48 / 52 Suricata 7.x Self-Audit & Perimeter Exposure Suite
# Scope: Inbound HTTP requests for private-key, keystore, and debug-log extensions (.key, .pem, .p12, .pfx, .log)
and outbound HTTP 200 responses serving PEM private keys, compiled LightSAML container caches, and binary PKCS#12
archives
# Validation: Static syntax and lint_suricata.py checked (0 problems); validate in staging before inline
enforcement.

alert http any any -> $HOME_NET any (msg:"ODIN R48 Inbound HTTP Request for Private Key File Extension";
flow:established,to_server; http.method; content:"GET"; bsize:3; http.uri; content:".key"; endswith; bsize:>=5;
fast_pattern; pcre:"/\/[a-z0-9_.-]+\.key$/i"; classtype:credential-theft; sid:9548001; rev:2; metadata:policy
security-ips, mitre_technique_id T1552.004;)
alert http any any -> $HOME_NET any (msg:"ODIN R48 Inbound HTTP Request for PEM Key or Certificate File
Extension"; flow:established,to_server; http.method; content:"GET"; bsize:3; http.uri; content:".pem"; endswith;
bsize:>=5; fast_pattern; pcre:"/\/[a-z0-9_.-]+\.pem$/i"; classtype:credential-theft; sid:9548002; rev:2;
metadata:policy security-ips, mitre_technique_id T1552.004;)
alert http $HOME_NET any -> any any (msg:"ODIN R48 Self-Audit Compiled SAML Credential Container Served Over
HTTP"; flow:established,to_client; http.stat_code; content:"200"; bsize:3; file.data; content:"Symfony\
\Component\\DependencyInjection"; content:"LightSaml\\Credential\\"; fast_pattern; classtype:policy-violation;
sid:9548003; rev:2; metadata:policy self_audit, mitre_technique_id T1606.002;)
alert http any any -> $HOME_NET any (msg:"ODIN R48 Inbound HTTP Request for PKCS12 P12 Keystore Extension";
flow:established,to_server; http.method; content:"GET"; bsize:3; http.uri; content:".p12"; endswith; bsize:>=5;
fast_pattern; pcre:"/\/[a-z0-9_.-]+\.p12$/i"; classtype:credential-theft; sid:9548004; rev:2; metadata:policy
security-ips, mitre_technique_id T1552.004;)
alert http any any -> $HOME_NET any (msg:"ODIN R48 Inbound HTTP Probe for Exposed PKCS12 PFX Private Keystore";
flow:established,to_server; http.method; content:"GET"; bsize:3; http.uri; content:".pfx"; endswith; bsize:>=5;
fast_pattern; pcre:"/\/[a-z0-9_.-]+\.pfx$/i"; classtype:credential-theft; sid:9548005; rev:1; metadata:policy
security-ips, mitre_technique_id T1552.004;)
alert http $HOME_NET any -> any any (msg:"ODIN R48 Self-Audit PEM Asymmetric Private Key Served Over HTTP";
flow:established,to_client; http.stat_code; content:"200"; bsize:3; file.data; content:"PRIVATE KEY-----";
fast_pattern; pcre:"/-----BEGIN (RSA |EC |DSA |OPENSSH |ENCRYPTED )?PRIVATE KEY-----/"; classtype:policy-
violation; sid:9548006; rev:1; metadata:policy self_audit, mitre_technique_id T1552.004;)
alert http $HOME_NET any -> any any (msg:"ODIN R48 Self-Audit Binary PKCS12 Keystore Served Over HTTP";
flow:established,to_client; http.stat_code; content:"200"; bsize:3; file.data; content:"|30 82|"; depth:2;
content:"|02 01 03 30|"; within:6; content:"|06 09 2A 86 48 86 F7 0D 01 07 01|"; fast_pattern; classtype:policy-
violation; sid:9548007; rev:1; metadata:policy self_audit, mitre_technique_id T1552.004;)
alert http any any -> $HOME_NET any (msg:"ODIN R48 Inbound HTTP Request for Application Debug Log Extension";
flow:established,to_server; http.method; content:"GET"; bsize:3; http.uri; content:".log"; endswith; bsize:>=5;
fast_pattern; pcre:"/\/[a-z0-9_.-]+\.log$/i"; classtype:attempted-recon; sid:9548008; rev:2; metadata:policy
security-ips, mitre_technique_id T1552.001;)
```

odin_exposed_signing_keys_saml_and_merchant_certs.yar

YARA

```
/*
   Odin by Brandefense - Monograph 48 / 52 Bounded YARA Self-Audit Suite
   Scope: Static posture scanning of web document roots, upload directories, and container artifacts for exposed
PEM RSA/EC signing keys, compiled Symfony LightSAML credential caches, binary PKCS#12 (.p12/.pfx) keystores, and
Laravel/Craft debug logs leaking bearer auth context.
*/

rule ODIN_Exposed_PEM_RSA_Or_EC_Private_Signing_Key {
    meta:
        author = "Odin by Brandefense"
        monograph = "48 / 52"
        description = "Detects PEM-encoded RSA, PKCS#8, or EC asymmetric private signing keys inside web-
accessible trees"
        mitre_attack = "T1552.004"
```

```
        date = "2026-11-21"
    strings:
        $hdr1 = "-----BEGIN RSA PRIVATE KEY-----" ascii
        $hdr2 = "-----BEGIN PRIVATE KEY-----" ascii
        $hdr3 = "-----BEGIN EC PRIVATE KEY-----" ascii
        $hdr4 = "-----BEGIN ENCRYPTED PRIVATE KEY-----" ascii
        $ftr1 = "-----END RSA PRIVATE KEY-----" ascii
        $ftr2 = "-----END PRIVATE KEY-----" ascii
        $ftr3 = "-----END EC PRIVATE KEY-----" ascii
        $ftr4 = "-----END ENCRYPTED PRIVATE KEY-----" ascii
        $asn1_b64_a = "MII" ascii
        $asn1_b64_b = "MHcCAQEE" ascii
        $asn1_b64_c = "MIGkAgEB" ascii
    condition:
        filesize < 64KB and (1 of ($hdr*)) and (1 of ($ftr*)) and (1 of ($asn1_b64_*))
}

rule ODIN_Compiled_Symfony_LightSAML_Credential_Store_Cache {
    meta:
        author = "Odin by Brandefense"
        monograph = "48 / 52"
        description = "Detects compiled Symfony DependencyInjection container LightSAML credential store or
resolver PHP service classes"
        mitre_attack = "T1606.002"
        date = "2026-11-21"
    strings:
        $cls1 = "LightSaml\\Credential\\" ascii
        $cls2 = "X509Credential" ascii
        $cls3 = "KeyHelper::createPrivateKey" ascii
        $di1 = "Symfony\\Component\\DependencyInjection" ascii
        $di2 = "Container" ascii
    condition:
        filesize < 256KB and (2 of ($cls*)) and $di1 and $di2
}

rule ODIN_PKCS12_Merchant_Or_TLS_Keystore_Bundle {
    meta:
        author = "Odin by Brandefense"
        monograph = "48 / 52"
        description = "Detects binary RFC 7292 PKCS#12 (.p12 and .pfx) merchant mTLS or TLS keystore bundles"
        mitre_attack = "T1552.004"
        date = "2026-11-21"
    strings:
        $p12_v3 = { 30 82 ?? ?? 02 01 03 30 82 }
        $oid_pkcs7_data = { 06 09 2A 86 48 86 F7 0D 01 07 01 }
        $oid_pkcs7_enc = { 06 09 2A 86 48 86 F7 0D 01 07 06 }
        $oid_pkcs12_bag = { 06 0B 2A 86 48 86 F7 0D 01 0C 0A 01 }
    condition:
        filesize < 128KB and $p12_v3 at 0 and $oid_pkcs7_data and ($oid_pkcs7_enc or $oid_pkcs12_bag)
}

rule ODIN_Exposed_Laravel_Or_Craft_Debug_Log_With_Auth_Context {
    meta:
        author = "Odin by Brandefense"
        monograph = "48 / 52"
        description = "Detects exposed Laravel or Craft/Yii2 framework debug log files containing stack traces
alongside bearer token or OAuth signing context"
        mitre_attack = "T1552.001"
        date = "2026-11-21"
    strings:
        $fw_laravel1 = ".ERROR:" ascii
        $fw_laravel2 = "[stacktrace]" ascii
        $fw_laravel3 = "Illuminate\\Foundation\\" ascii
        $fw_craft1 = "yii\\web\\" ascii
        $fw_craft2 = "craft\\web\\Application" ascii
        $auth1 = "Authorization" ascii
        $auth2 = "Bearer eyJ" ascii
        $auth3 = "League\\OAuth2\\Server" ascii
    condition:
        filesize < 50MB and ((2 of ($fw_laravel*)) or (1 of ($fw_craft*))) and (1 of ($auth*))
}
```

sentinel_exposed_signing_keys_and_saml_cache_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 48 / 52 Microsoft Sentinel & Defender XDR Self-Audit Hunt Query
// Purpose: Correlate ad-hoc HTTP directory listeners or key-generation/permission commands exposing OAuth2, JWT,
SAML, or merchant mTLS private keys with open web sockets
let Lookback = 7d;
let ApprovedKeyBuildHosts = dynamic([
    "pki-vault-builder-01.corp.example",
    "ci-artifact-signer-01.corp.example"
]);
let KeyExposureProcesses =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedKeyBuildHosts))
    | where FileName in~ ("python", "python3", "php", "openssl", "chmod", "cp", "mv", "tar", "nginx", "apache2",
"httpd")
    | where ProcessCommandLine has_any (
        "http.server",
        "SimpleHTTPServer",
        "passport:keys",
        "lexik:jwt:generate-keypair",
        "pkcs12 -export",
        "genrsa",
        "ecparam",
        ".key",
        ".pem",
        ".p12",
        ".pfx",
        ".log"
    )
    | extend ExposureCategory = case(
        ProcessCommandLine has_any ("passport:keys", "lexik:jwt:generate-keypair", "genrsa", "ecparam", ".key",
".pem"), "OAuth2OrJWTSigningKey",
        ProcessCommandLine has_any ("pkcs12 -export", ".p12", ".pfx"), "MerchantOrTLSPKCS12Keystore",
        ProcessCommandLine has_any ("http.server", "SimpleHTTPServer"), "AdHocHTTPDirectoryListener",
        "FrameworkDebugLogOrKeyCopy"
    )
    | project ProcTime = Timestamp, DeviceId, DeviceName, AccountName, FileName, ProcessCommandLine,
ExposureCategory;
let WebListenerActivity =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedKeyBuildHosts))
    | where ActionType in ("ListeningConnectionCreated", "InboundConnectionAccepted")
    | where LocalPort in (80, 443, 8000, 8080, 8443)
    | summarize
        InboundEvents = count(),
        ListeningPorts = make_set(LocalPort, 10),
        RemoteCallers = dcount(RemoteIP),
        LastNetTime = max(Timestamp)
      by DeviceId, DeviceName;
KeyExposureProcesses
| join kind=leftouter WebListenerActivity on DeviceId, DeviceName
| extend RemediationAction = case(
    ExposureCategory in ("OAuth2OrJWTSigningKey", "MerchantOrTLSPKCS12Keystore") and array_length(ListeningPorts)
> 0, "URGENT_CLOSE_WEB_PATH_AND_ROTATE_SIGNING_KEY",
    ExposureCategory == "AdHocHTTPDirectoryListener" and array_length(ListeningPorts) > 0,
"URGENT_TERMINATE_HTTP_LISTENER_AND_AUDIT_KEYS",
    "AUDIT_KEY_STORAGE_OUTSIDE_WEB_ROOT"
)
| project
    ProcTime,
    LastNetTime,
    DeviceName,
    AccountName,
    FileName,
    ExposureCategory,
    InboundEvents = coalesce(InboundEvents, 0),
```

```
    RemoteCallers = coalesce(RemoteCallers, 0),
    ListeningPorts,
    RemediationAction,
    ProcessCommandLine
| sort by InboundEvents desc, ProcTime desc
```

sigma_exposed_oauth_jwt_saml_and_merchant_keys.yml

SIGMA

```
title: Web Server Serving OAuth2 JWT SAML Signing Keys or Merchant mTLS Bundles
id: 7c48a1b2-93d4-4e61-a801-954800100001
status: experimental
description: Detects HTTP 200 or 206 responses serving files with private-key, PKCS12 keystore, certificate, or
debug-log extensions (.key, .pem, .p12, .pfx, .log) from a web server.
author: Odin by Brandefense
date: 2026-11-21
references:
  - https://attack.mitre.org/techniques/T1552/004/
  - https://attack.mitre.org/techniques/T1606/002/
logsource:
  category: webserver
detection:
  selection_status:
    sc-status:
      - 200
      - 206
  selection_uri:
    cs-uri-stem|endswith:
      - '.key'
      - '.pem'
      - '.p12'
      - '.pfx'
      - '.log'
  condition: selection_status and selection_uri
falsepositives:
  - Publicly served X.509 certificate (.pem) downloads or test fixtures; private signing keys, PKCS12 keystores,
and debug logs must never be served over HTTP.
level: critical
tags:
  - attack.credential-access
  - attack.t1552.004
  - attack.t1606.002
---
title: Ad-Hoc HTTP Directory Server or Permissive Mode Change on Signing Keys
id: 7c48a1b2-93d4-4e61-a801-954800100002
status: experimental
description: Detects ad-hoc Python or PHP HTTP directory servers or chmod invocations referencing key or PKCS12
keystore extensions (.key, .pem, .p12, .pfx) on Linux application servers.
author: Odin by Brandefense
date: 2026-11-21
references:
  - https://attack.mitre.org/techniques/T1552/004/
logsource:
  product: linux
  category: process_creation
detection:
  selection_http_listener:
    Image|endswith:
      - '/python'
      - '/python3'
      - '/php'
    CommandLine|contains:
      - '-m http.server'
      - 'SimpleHTTPServer'
      - '-S 0.0.0.0:'
  selection_key_chmod:
    Image|endswith:
      - '/chmod'
    CommandLine|contains:
```

```
      - '.key'
      - '.pem'
      - '.p12'
      - '.pfx'
  filter_approved_hosts:
    ComputerName:
      - 'pki-vault-builder-01.corp.example'
      - 'ci-artifact-signer-01.corp.example'
  condition: (selection_http_listener or selection_key_chmod) and not filter_approved_hosts
falsepositives:
  - Dedicated PKI build or CI signing runners explicitly listed in filter_approved_hosts.
level: high
tags:
  - attack.credential-access
  - attack.t1552.004
---
title: Signing Key or Merchant PKCS12 Bundle Written Inside Public Web Root
id: 7c48a1b2-93d4-4e61-a801-954800100003
status: experimental
description: Detects creation of private key or PKCS12 keystore files (.key, .pem, .p12, .pfx) directly inside
web server document roots (/var/www/, /public_html/, /htdocs/).
author: Odin by Brandefense
date: 2026-11-21
references:
  - https://attack.mitre.org/techniques/T1552/004/
  - https://www.rfc-editor.org/rfc/rfc7519
logsource:
  product: linux
  category: file_event
detection:
  selection_public_dir:
    TargetFilename|contains:
      - '/var/www/'
      - '/public_html/'
      - '/htdocs/'
  selection_key_ext:
    TargetFilename|endswith:
      - '.key'
      - '.pem'
      - '.p12'
      - '.pfx'
  filter_approved_hosts:
    ComputerName:
      - 'pki-vault-builder-01.corp.example'
      - 'ci-artifact-signer-01.corp.example'
  condition: selection_public_dir and selection_key_ext and not filter_approved_hosts
falsepositives:
  - None expected inside public document roots; store keys outside document root with 0600 permissions.
level: high
tags:
  - attack.credential-access
  - attack.t1552.004
```

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/exposed-oauth2-jwt-saml-and-merchant-keys/
Short Read · brandefense.io/odin-research/blog/how-misconfigured-web-roots-leak-oauth2-jwt-saml-and-merchant/
Defense Kit · brandefense.io/odin-research/defense-kit/