ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 09 / 30 · DEVELOPER TOOLING

Electron IDE Secret Theft and Single-Host Exploit Workbenches

How attackers mine desktop AI tools for OAuth secrets, and why most "shared C2 hubs" are look-alikes

Prepared November 2026
Full Report · Public edition · Version 1.1

2,019
Unique C2, callback and credential values found across 140 exposed servers (7,146 file hits)

52
Values that looked like shared hubs on 3 or more servers: 40 open-source fixtures and 12 public constants

1,544
Values that remain real findings (76.5%) once fixtures and constants are set aside

126
Real values seen on exactly 2 servers, the widest real overlap; 8 sit on adversary hosts

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

How attackers mine desktop AI tools for OAuth secrets, and why most "shared C2 hubs" are look-alikes

01	Executive Summary	04
02	Look-Alike Hubs: What 2,019 Shared Values Really Were	07
03	Electron IDE Secret Extraction and CLI Token Fallback	11
04	Java Deserialization and Recon Workbenches	15
05	Rogue JWKS Keys and Cloud Metadata Redirectors	19
06	Other Adversary Staging Hosts	22
07	What Defenders Should Do	24
08	Detection Rules	27
09	Indicators	30
10	Scope and Limits	33
11	Sources	35
A	Indicators of compromise (27)	37
B	Detection rules (4 files)	39

IN SHORT

In data with an evidence cutoff of 15 November 2026, Odin by Brandefense found a 24-script workbench built to pull OAuth secrets out of a commercial Electron AI IDE. We also found three single-host Java and JWKS exploit kits. Across 2,019 values shared between 140 exposed servers, all 52 apparent multi-server hubs were open-source fixtures or public constants. Replace static desktop OAuth secrets with PKCE, pin JWKS origins, and check multi-host matches before blocking them.

01

CHAPTER 01

Executive Summary

IN SHORT

In data with an evidence cutoff of 15 November 2026, Odin by Brandefense found a 24-script workbench built to pull OAuth secrets out of a commercial Electron AI IDE. We also found three single-host Java and JWKS exploit kits. Across 2,019 values shared between 140 exposed servers, all 52 apparent multi-server hubs were open-source fixtures or public constants. Replace static desktop OAuth secrets with PKCE, pin JWKS origins, and check multi-host matches before blocking them.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Desktop AI coding assistants and local agent gateways keep powerful credentials on developer machines. These include OAuth client secrets, model API keys and cloud session tokens. This report looks at three things: staged tooling that targets those secrets, three compact exploit workbenches, and a common analysis trap that makes unrelated servers look like one C2 network.

1. **A 24-script workbench targets OAuth secrets inside an Electron AI IDE.**
   Host W (US, address withheld; 1,228 files) holds 24 Node.js scripts written to search a 249 MB Electron archive and a 12.5 MB CLI bundle. No recovered IDE secret was found. Separately, two exposed developer gateways serve `.env` files and upstream code that can fall back to CLI or Keychain tokens.

2. **Three single-host workbenches stage Java deserialization, API recon and rogue JWKS kits.**
   Host 202.162.109[.]5 stages rogue LDAP and MySQL servers, an H2 database payload and a reverse shell to 143.92.57[.]247. Host 129.226.71[.]134 holds 15 custom Python recon tools. Host 38.246.253[.]11 stages a rogue RS256 JWKS keyset and cloud metadata redirectors.

3. **All 52 apparent shared C2 hubs were fixtures or public constants.**
   Of 2,019 values, 52 appeared on three or more servers, up to 71. Forty were bundled open-source fixtures and 12 were public constants such as cloud metadata URLs. No real value appeared on more than 2 servers.

Three Decisions for Leaders

1. **Remove static OAuth secrets from desktop apps.** Replace embedded client secrets in Electron archives and companion CLI bundles with PKCE (RFC 8252). Turn on Electron ASAR integrity checks.
2. **Pin JWKS origins and harden cloud metadata.** Accept token-signing keys only from allowlisted origins. Re-check private and link-local addresses on every HTTP redirect hop, and turn on each cloud provider's metadata protections.
3. **Check multi-host matches before you block.** Bundled wordlists, documentation URLs and cloud metadata addresses appear on many servers. Filter them out of threat feeds so they never look like a shared C2 network.

The Study in Numbers

Number	What it is
2,019	Unique C2, callback and credential values found across 140 exposed servers (7,146 file hits)
52	Values that looked like shared hubs on 3 or more servers: 40 open-source fixtures and 12 public constants
1,544	Values that remain real findings (76.5%) once fixtures and constants are set aside
126	Real values seen on exactly 2 servers, the widest real overlap; 8 sit on adversary hosts
24	Node.js scripts written to slice a 249 MB Electron IDE archive on Host W
192 of 200	AI provider API keys that are real exposures rather than test fixtures
27	Published indicators: 12 IPv4 hosts and 15 SHA-256 file hashes

How to use this report.

Security leaders can read Chapters 1, 7 and 10. Detection engineers and SOC analysts should start with Chapters 7 to 9. CTI analysts will find the workbench detail in Chapters 3 to 6 and the hub analysis in Chapter 2.

02

CHAPTER 02

Look-Alike Hubs: What 2,019 Shared Values Really Were

IN SHORT

Values that appear on dozens of servers look like shared attacker infrastructure. In this data set, every one of them turned out to be a bundled fixture or a public constant.

INSIDE THIS CHAPTER

The trap 07

What the values turned out to be 08

Where the fake secrets came from 08

The 12 public constants behind the biggest "hubs" 08

The real overlap is small 09

02 Look-Alike Hubs: What 2,019 Shared Values Really Were

The trap

A common way to find shared attacker infrastructure is to match the same URL, key or callback across many exposed servers. Taken at face value, our data looked alarming. Of 2,019 unique values, 210 appeared on two or more servers. Fifty-two appeared on 3 to 71 servers each.

Raw counts also suggested wide secret exposure: 66 Google API keys, 42 private keys and 30 AWS access keys. Most of these were not what they seemed.

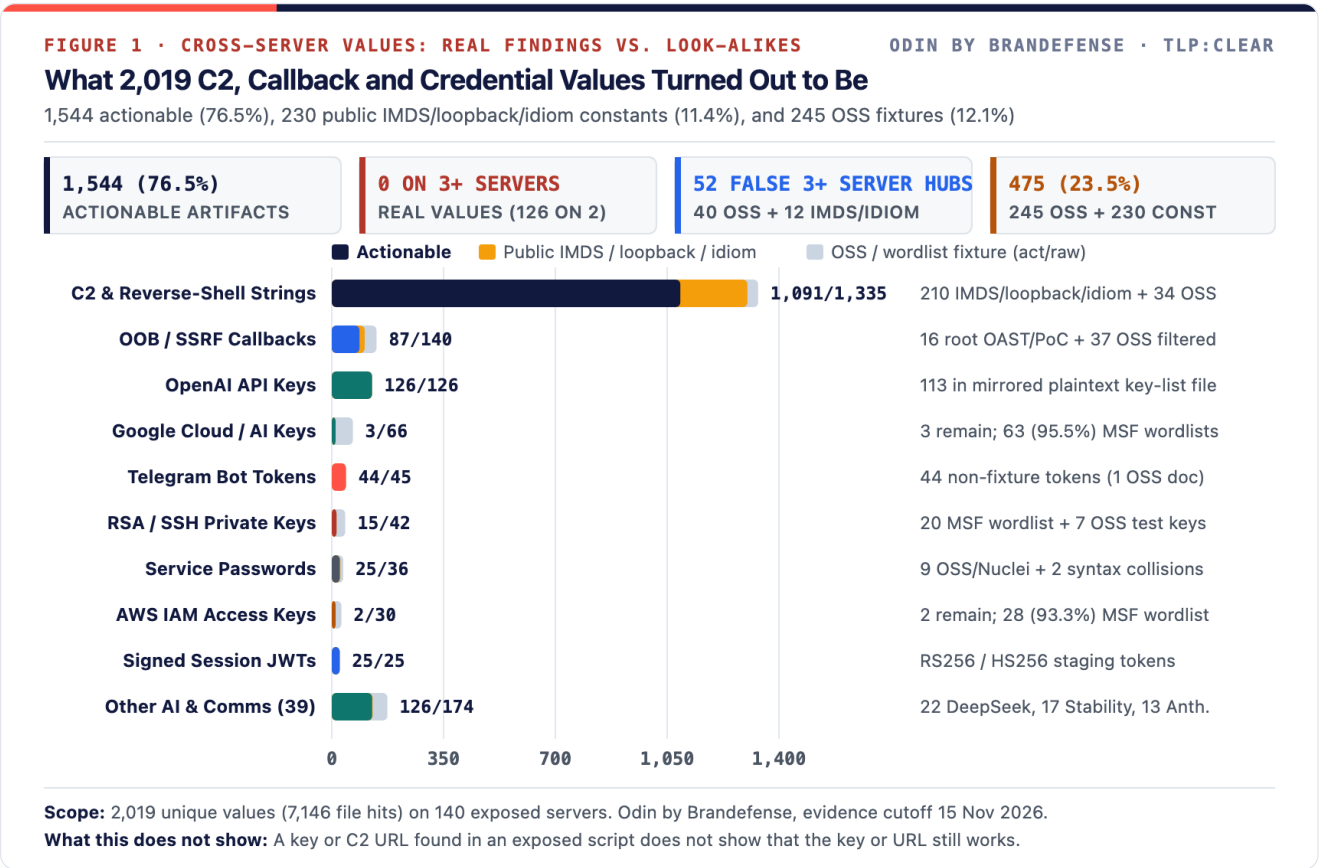


Figure 1: What 2,019 cross-server C2, callback and credential values turned out to be

In Figure 1, the top bar adds the one bare reverse-shell prefix to the 1,334 C2 URLs, so it shows 1,335.

What the values turned out to be

Outcome	Values	Share	What it covers
Real findings	1,544	76.5%	Includes staged C2 URLs (1,091), AI provider keys (192), private keys and signed JWTs (40), and per-session OAST and Telegram tokens
Open-source fixtures	245	12.1%	Documentation and unit-test examples, security wordlists, cloned Nuclei templates
Public constants	230	11.4%	Loopback and RFC1918 URLs (200), public cloud metadata URLs (9), root OAST nameservers and code-syntax matches (20), one bare shell prefix
Total	2,019	100%	

The two largest look-alike clusters were a Jupyter live-kernel documentation URL on 71 servers and a `cli-config.yaml.example` endpoint on 67 servers. Both ship inside open-source packages.

Where the fake secrets came from

A single Metasploit wordlist, `flask_secret_keys.txt`, explains most of the alarming secret counts.

Secret type	Found	Real	Main source of the rest
C2 URLs	1,334	1,091	200 loopback or RFC1918 URLs and 9 public cloud metadata URLs
Out-of-band callbacks	140	87	Nuclei CVE templates (37) and root OAST domains (16)
OpenAI API keys	126	126	None; 113 sit in one mirrored plaintext key list
Google API keys	66	3	63 in Metasploit `flask_secret_keys.txt`
Telegram bot tokens	45	44	1 open-source documentation example
Private keys	42	15	20 in the Metasploit wordlist, 7 open-source test keys
Generic passwords	36	25	`aiohttp` and `yarl` tests, plus `fallback_getpass` on 7 servers
AWS access keys	30	2	28 in the Metasploit wordlist
Slack tokens	13	1	12 in MobSF signature rules and open-source tests

Other real AI keys include 22 DeepSeek, 17 Stability, 13 Anthropic, 7 OpenRouter and 3 Groq keys. We did not test whether any key still works.

The 12 public constants behind the biggest "hubs"

Twelve of the 52 wide clusters were not open-source files at all. Each is a public value that many unrelated servers contain.

Servers	Value	Example host and file
42	`bash -i >& /dev/tcp/` (bare shell prefix, no target)	202.162.109[.]5, `/hr.sh`
42	`http://169.254.169.254/latest/meta-data/` (AWS metadata)	129.226.71[.]134, `/vt2.py`
23	`http://169.254.169.254/metadata/instance` (Azure metadata)	Developer host, withheld

Servers	Value	Example host and file
21	`http://169.254.169.254/latest/meta-data/instance-id`	Developer host, withheld
19	`http://100.100.100.200/latest/meta-data/` (Alibaba Cloud metadata)	38.246.253[.]11, `/redir.log`
15	`http://169.254.169.254/metadata/v1/id` (DigitalOcean metadata)	Developer host, withheld
7	`fallback_getpass` (code identifier)	Developer gateway, withheld
5	`http://127.0.0.1:18789/__openclaw__/canvas/` (loopback)	Developer gateway, withheld
5	`http://169.254.169.254/latest/user-data/`	8.137.114[.]242, `generate_rules.py`
4	`keyboardshortcuts` (code identifier)	Unrelated host, withheld
3	`http://100.100.100.200/latest/meta-data/ram/security-credentials/`	38.246.253[.]11, `/redir.log`
3	`e.oauth.grantType?n` (minified code fragment)	8.138.12[.]59, `/k4_app.js`

The top two rows matter most. Each made a single-host workbench look like part of a 42-server network. Chapter 4 shows what those hosts really are.

The real overlap is small

Among real findings, no value appears on more than 2 servers. There are 126 such two-server values, anchored to 6 servers.

Anchor host	Port	Values shared with one other server	Context
Exposure host, withheld	8000	113	Plaintext list of OpenAI project keys; owner unknown
43.230.171[.]150	8080	7	`/apply_AC2.json`, `/D_job.sql`, `/AD_svc.sql` and a settings file; adversary host
Developer host, withheld	4000	3	Agent session logs with 1 OpenAI, 1 Anthropic and 1 Telegram token
129.226.71[.]134	8899	1	Google API key in a staged web template; adversary host
Developer host, withheld	8899	1	DeepSeek key in an agent `.env` file
Developer host, withheld	8888	1	OpenRouter key in an agent profile `.env` file

Only 8 of the 126 values sit on the 2 adversary hosts. The other 118 sit on 4 hosts that look like exposed developer or unattributed machines. Those 4 hosts are not on any block list in this report.

CHAPTER 03

Electron IDE Secret Extraction and CLI Token Fallback

IN SHORT

One exposed server holds 24 scripts written to dig OAuth secrets out of a commercial Electron AI IDE. Separately, exposed agent gateways ship code that can borrow a developer's CLI login.

INSIDE THIS CHAPTER

Why Electron apps leak secrets 11

The 24-script workbench on Host W 11

What else sits on the same server 12

Agent gateways that can borrow CLI logins 12

03 Electron IDE Secret Extraction and CLI Token Fallback

Why Electron apps leak secrets

Electron packs an app's source into an `app.asar` file. That file is not encrypted. Any Node.js script can read it with ordinary file calls. RFC 8252 (OAuth 2.0 for Native Apps) states that a desktop binary cannot keep a static client secret confidential.

So if an app embeds `OAUTH_APP_SECRET` or `appSecret` inside `app.asar`, a local script can recover it. The holder of that secret can then act as the app in the OAuth `authorization_code` exchange.

The 24-script workbench on Host W

Host W (US, address withheld) exposes 1,228 files totalling 3,066,253 bytes. They show someone reverse-engineering a commercial Electron AI IDE, whose name we withhold.

Instead of unpacking the archive with the standard `@electron/asar` tool, the operator wrote 24 numbered scripts. They run from `find_secret.js` through `find_secret24.js`, each between 303 and 1,148 bytes. The scripts load the IDE's `resources/app.asar` (about 249 MB) and a companion CLI `.cjs` bundle (about 12.5 MB) as text.



Figure 2: Iterative ASAR secret extraction on Host W and CLI token fallback in a developer gateway

The scripts move through three phases, from simple string search to slicing the bundled code.

Phase	Count	What the scripts look for
1. String and login search	9	`OAUTH_APP_SECRET`, `OAUTH_CLIENT_SECRET`, `appSecret` and `tokenByAuthCode`, then walk back to the enclosing function to map the login call chain
2. Child-process environment	8	The process spawner (`spawnHostProcess`) and its environment object (`hostProcessLocalEnv`), to find `.env` loaders and injected secrets
3. Bundle slicing	7	Sub-megabyte windows of `app.asar` around `hostProcessLocalEnv`, measuring distance between webpack chunks to isolate the function that returns the secret

The target archive and any extracted secret were not on the server. The scripts show intent and method, not a recovered secret or a breach of the IDE vendor.

What else sits on the same server

Tool	Files	Purpose
Panel automation	`agent_panel.js`, `agent_v3.js`, `agent_v4.js`	Playwright scripts that drive a web panel
Browser capture	`cdp_screenshot.js`	Chrome DevTools Protocol screenshot utility
Account registration	`exploit_register.py`, `exploit_semi.py`, `exploit_v2.py` to `exploit_v15.py` (11 scripts)	Iterative scripts that register accounts
Wallet keystore	One file, 2,038 bytes (path withheld)	Exposed cryptocurrency wallet keystore

A dual-use caveat. The registration scripts create accounts with names that suggest a penetration test. This server may belong to a red team that left its test box exposed, or to a hostile operator. We cannot tell which, and authorization is unknown.

Agent gateways that can borrow CLI logins

Two exposed gateways run the open-source NousResearch `hermes-agent` on port 8080. Their owners appear to be developers, not attackers, so both hosts stay anonymous.

The larger gateway serves 523 files over plain HTTP. They include its live `/.hermes/.env` (26,669 bytes, 542 lines) and `gateway.log` (17,889 bytes). They also include the upstream module `anthropic_credentials.py` (41,534 bytes, 816 lines).

That module's `resolve_anthropic_token()` function tries four sources in order.

Step	Source	What it reads
1	Environment token	`ANTHROPIC_TOKEN` or `CLAUDE_CODE_OAUTH_TOKEN`
2	Static API key	`ANTHROPIC_API_KEY` (`sk-ant-api*` keys, sent as `x-api-key`)
3	Hermes OAuth file	`~/.hermes/.anthropic_oauth.json` (`sk-ant-oat*` tokens, sent as a Bearer token)
4	Borrowed CLI login	`~/.claude/.credentials.json` or the macOS Keychain entry for Claude Code, copied into the Hermes credential pool (`auth.json`)

Step 4 is an upstream feature, not malware. The risk comes from serving `~/.hermes/` over HTTP. If a borrowed token or a session history database is stored there, anyone can read it.

We found no `auth.json`, no CLI or Keychain token, no SQLite history and no log showing step 4 ran. Across all servers in the study, `.env` files expose 22 DeepSeek and 13 Anthropic keys.

CHAPTER 04

Java Deserialization and Recon Workbenches

IN SHORT

Two hosts looked like members of a 42-server network only because each contains a common public string. Each is in fact a self-contained, single-host workbench.

INSIDE THIS CHAPTER

202.162.109[.]5: a Java deserialization kit 16

129.226.71[.]134: an API and cloud recon workbench 17

04 Java Deserialization and Recon Workbenches

Hosts 202.162.109[.]5 and 129.226.71[.]134 each appeared in a 42-server cluster. For the first, the shared value was the bare shell prefix `bash -i >& /dev/tcp/`. For the second, it was the AWS metadata URL. Both strings appear on many unrelated servers, so neither cluster is evidence of shared infrastructure.

FIGURE 3 · JAVA JNDI/JDBC & RECON WORKBENCHES — ODIN BY BRANDEFENSE · TLP:CLEAR

Multi-Protocol Java JNDI/JDBC Kit (202.162.109[.]5) & 390-Entry Recon Host (129.226.71[.]134)

Single-host offensive toolkits whose 42-server matches traced to bash -i & AWS IMDS constants

WORKBENCH A · 202.162.109[.]5:8081 (23 FILES, 15 SRC)

Multi-Protocol Java & JDBC Deserialization Kit

1. Rogue LDAP (/evil-ldap.py · 3,698 B)
Serves BER javaSerializedData on TCP 1389
Bypasses JDK 8u191+ trustURLCodebase=false

2. Dual Rogue MySQL (3,326 B & 3,630 B)
evil-mysql-blob.py (:3312) & S.py (:3314)
BLOB resultset for Connector/J autoDeserialize

3. H2 JDBC RCE (/evil1.sql · 505 B)
CREATE ALIAS HR_X inline Java Runtime.exec
Written to send web-root listing -> 143.92.57[.]247

4. Dropper & Public Constant (/hr.sh · 56 B)
Targets external /dev/tcp/143.92.57[.]247/4444 listener
42 servers share the common string 'bash -i >& /dev/tcp/'

WORKBENCH B · 129.226.71[.]134:8899 (390 DIR, 15+1 PY)

Cloud, API & DNS Reconnaissance Workbench

1. Staged IMDS Probe (/vt2.py · 1,444 B)
Written for NoSQLi/SQLi/SSTI & AWS IMDS probing
42 servers share the common metadata URL

2. Staged API & Bypass Scripts (apienum, byp)
Designed for REST route discovery & 403 testing
Paired with actenum.py & cffi.py TLS spoofing

3. Staged DNS & Mobile Tools (dnsbrute, mob)
Configured for subdomain & mobile API probing
15 custom Python tools (+1 setup.py); no exec log

4. Real Values on 2 Servers (126 on 6 hosts)
118 unattributed/dev exposed keys (113 on one host)
8 on adversary hosts (43.230.171[.]150: 7; 129.226.71[.]134: 1)

Scope: Files retrieved from both hosts. Odin by Brandefense, evidence cutoff 15 Nov 2026.
What this does not show: httpd.log on 202.162.109[.]5 records 127.0.0.1 self-test fetches of evil1.sql/x.js, not use against a victim.

Figure 3: Single-host Java JNDI/JDBC and recon workbenches behind two 42-server look-alike matches

202.162.109[.]15: a Java deserialization kit

This host (Singapore) exposes 23 directory entries across ports 8081 and 8080. We retrieved 15 files. The operator did not use public frameworks such as `ysoserial` or `rogue_mysql_server`. Instead, they wrote small Python emulators for each protocol.

File	Size	Listens on	What it does
`/evil-ldap.py`	3,698 B, 103 lines	TCP 1389	Rogue JNDI LDAP server. Answers a search request with `javaClassName: Exploit` and a `javaSerializedData` payload, which sidesteps the `trustURLCodebase` limit in JDK 8u191 and later
`/evil-mysql-blob.py`	3,326 B, 105 lines	TCP 3312	Rogue MySQL server with banner `5.7.36-evilblob`. Returns a serialized object as a BLOB column to any MySQL Connector/J client using `autoDeserialize=true`
`/S.py`	3,630 B, 103 lines	TCP 3314	Configurable rogue MySQL server with banner `5.7.36-debug`. Lets the operator set column type and other fields, and injects a payload into `SELECT` results
`/evil1.sql`	505 B, 2 lines	Served over HTTP	H2 database script that defines `CREATE ALIAS HR_X` to call `Runtime.getRuntime().exec`, and sends `id`, `hostname` and a web-root listing to 143.92.57[.]247:8081
`/hr.sh`	56 B	Served over HTTP	Bash reverse shell to 143.92.57[.]247:4444
`/x.js`	172 B	Served over HTTP	Script that calls back to 143.92.57[.]247:8080
`/start-blob.sh`, `/start-s.sh`	236 B, 179 B		Launchers for the two MySQL servers

The server's own logs add context. `/ldap.log` (40,432 bytes, 1,472 lines) records `evil-ldap on :1389 payload=1323B` and repeated inbound LDAP connections. `/blob3312.log` (177 bytes) records two outside JDBC clients receiving 1,298-byte BLOB payloads.

`/httpd.log` (262,144 bytes, 3,287 lines) carries entries dated 25 October 2026. These show `GET /evil1.sql` requests from `127.0.0.1`, which are the operator's own self-tests. Internet scanner probes such as `GET /api/mcp` also appear. This date is a timestamp in the attacker's log file, not part of our observation period.

The operator's infrastructure spans 2 hosts: 202.162.109[.]15 and the callback host 143.92.57[.]247.

129.226.71[.]134: an API and cloud recon workbench

This host (Hong Kong, port 8899) exposes 390 directory entries in 106 subfolders, totalling 4,416,662 bytes. We retrieved 15 custom Python tools, plus 1 vendored `setup.py` from the open-source SendSafely client library. A staged RSA private key also sits on the server (path withheld).

Script	Size	What it is written to do
`/vt2.py`	1,444 B, 28 lines	Send 7 injection tests to `/security/v1/verify`: NoSQL injection, SQL injection, template injection (`{{7*7}}`), path traversal and an AWS metadata URL
`/apienum.py`	1,925 B, 27 lines	Probe 35 REST routes such as `/api_gateway/v1/users`, `/wallets`, `/kyc` and `/auth/v1/token` with 4 threads. Companions: `/enum2.py`, `/enum3.py`
`/actenum.py`	1,087 B, 14 lines	Enumerate 26 activity, admin, internal and debug endpoints with an Android Pixel 8 user agent
`/byp.py`	1,571 B, 23 lines	Probe 45 gateway routes with custom `X-platform` headers to find WAF bypass responses. Companions: `/probe.py`, `/leads.py`
`/cffi.py`	1,024 B, 20 lines	Test TLS-fingerprint bypass with `curl_cffi` impersonating Chrome 131 through 24 SOCKS5 proxies. Companions: `/geo.py`, `/mob.py`, `/mob2.py`, `/proxycheck.py`, `/pw2.py`
`/dnsbrute.py`	1,234 B, 31 lines	Brute-force subdomains with `dnspython` across 60 threads against `8.8.8[.]8` and `1.1.1[.]1`

The routes in these scripts (wallets, KYC, auth tokens) suggest interest in financial or crypto platforms. We found no execution log on this host, so we cannot say whether any target was probed.

CHAPTER 05

Rogue JWKS Keys and Cloud Metadata Redirectors

IN SHORT

Host 38.246.253[.]11 stages everything needed to test forged-token acceptance through the JWT `jku` header. It also runs redirectors aimed at Alibaba Cloud's metadata service.

INSIDE THIS CHAPTER

Path 1: a rogue JWKS keyset 19

Path 2: redirectors that reach cloud metadata 20

05 Rogue JWKS Keys and Cloud Metadata Redirectors

Host 38.246.253[.]11 (Hong Kong) exposes 32 directory entries across ports 8082 and 8085. We retrieved 9 files. Together they cover two attack paths.



Figure 4: Single-host workbench 38.246.253[.]11 pairs RS256 JWKS staging with cloud metadata redirectors

Path 1: a rogue JWKS keyset

A signed JWT can carry a `jku` header (RFC 7515). It points to a URL that serves the public keys (a JWKS, RFC 7517) used to check the signature. If a server trusts any `jku` origin, an attacker can sign a token with their own key and point `jku` at their own keyset.

The host stages all three parts of this attack.

Part	File	Size	Port
Attacker RSA private key	Name withheld	1,704 B	8082
Matching public key	`/jwt_pub.pem`	451 B (2048-bit RSA)	8085
Served keyset	`/jwks.json`	438 B, `"kid": "hktest1"`, `"alg": "RS256"`	8085

The access log (`/access.log`, 262,144 bytes, 1,183 lines) mostly records internet scanners and directory browsing. We therefore do not claim that any service accepted a forged token. The key ID `hktest1` is the most useful hunting value.

Path 2: redirectors that reach cloud metadata

Many web apps and AI agent sandboxes block requests to `169.254.169.254` or `100.100.100.200`. Many do not re-check the address after following an HTTP redirect. The host stages two small Python servers built on `BaseHTTPRequestHandler` to exploit that gap.

`/redir_server.py` (897 bytes, 23 lines) listens on TCP 8083 and answers six short paths with `302 Found`.

Path	Redirects to
`/r-meta`	`http://100.100.100.200/latest/meta-data/` (Alibaba Cloud ECS metadata root)
`/r-ram`	`http://100.100.100.200/latest/meta-data/ram/security-credentials/` (RAM role temporary credentials)
`/r-img`	`http://38.246.253[.]11:8082/test.png` (harmless test image)
`/r-local80`, `/r-local8080`	`http://127.0.0.1/` and `http://127.0.0.1:8080/`
`/r-self`	The host's own address (withheld)

`/redir2.py` (826 bytes, 21 lines) listens on TCP 8084. It maps `/r301`, `/r302`, `/r307` and `/r308` to redirects of the same status code. This tests which codes a client follows. Codes 307 and 308 also keep the `POST` method and body.

Two logs show the redirectors in use. `/redir.log` (112,905 bytes, 2,211 lines) and `/redir2.log` (14,456 bytes, 295 lines) record repeated 302 answers for `/r-ram` and `/r-meta`. The logs do not show which clients made the requests or what they received.

The metadata URLs inside `/redir.log` also produced two look-alike clusters, on 19 and 3 servers. Both are public constants (see Chapter 2).

CHAPTER 06

Other Adversary Staging Hosts

IN SHORT

Eight more hosts stage droppers, phishing kits, callback harvesters and exploit tooling. They complete the published list of 12 adversary IPv4 hosts.

06 Other Adversary Staging Hosts

Host	Port	What it stages
43.230.171[.]150	8080	`/AD_svc.sql`, `/D_job.sql`, `/apply_AC2.json` and a settings file; 7 of its values also appear on one other server (16 values in total)
38.207.179[.]179	8088	`/cmds.txt` and token files yielding 716 real C2 URL and token values
51.194.215[.]35	8080	`/b/*.sh` dropper scripts that fetch from `http://51.194.215[.]35:8080/h/*` (236 real values)
8.137.114[.]242	8080	GoPhish and recon tooling: `/gophish/gophish.log`, `/gophish/nohup.out`, `/ApolloFish/nohup.out` and key and config files (60 real values)
144.172.99[.]199	8080	`/assemble_dnslog.py` and `/dbg18.log` with 43 unique per-session OAST and DNSLog callbacks
8.138.12[.]59	9999	Fastjson tooling: `/fastjson_probe.py`, `/ssrf_nacos.sh`, `/ssrf_probe2.sh` and token-capture files (13 real values of 18)
38.60.227[.]210	8082	`/deploy_ghc2.sh` (a GitHub-based C2 agent), `/actor.ps1` and `/miner/install_user.sh` (8 real C2 URLs)
47.109.23[.]77	8888	`/jndi.log` recording `exploit.class` and downstream botnet payload URLs (8 real values)

Apart from the 7 values that 43.230.171[.]150 shares, nothing in this data links these hosts to each other or to the three workbenches. Treat them as separate operations.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

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

07 What Defenders Should Do

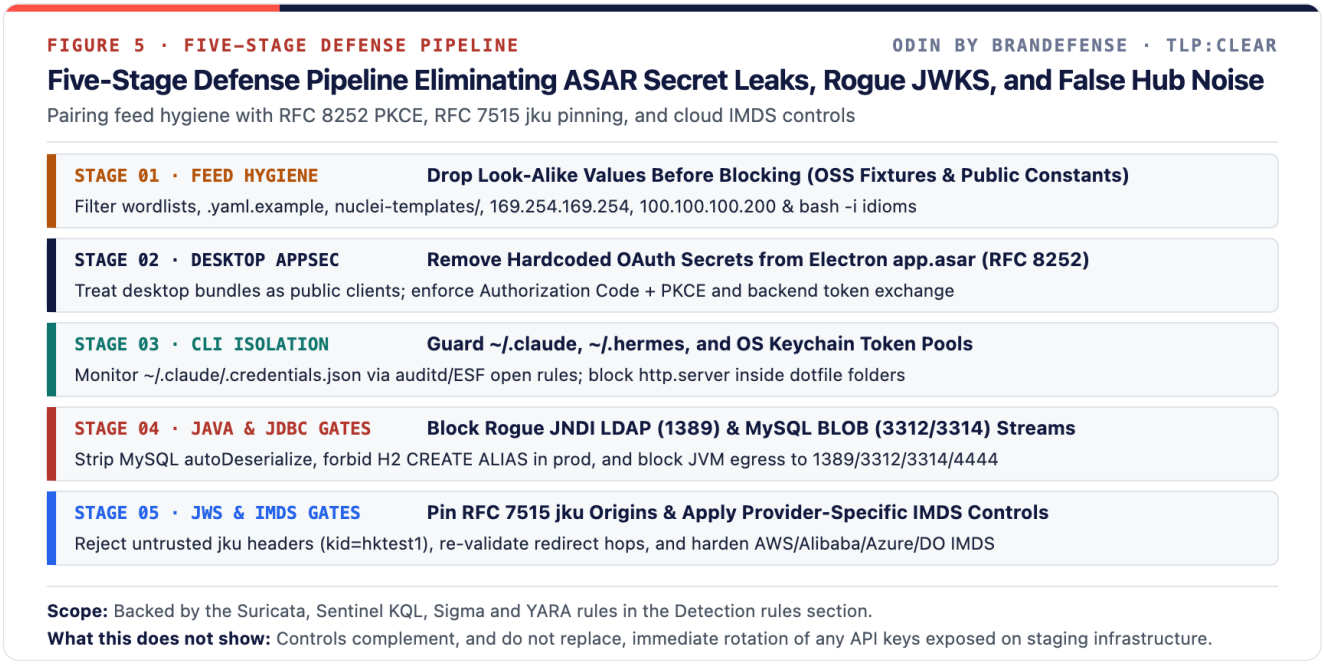


Figure 5: Five-stage defense pipeline for desktop OAuth secrets, Java and JWKS kits, and look-alike hubs

Within 24 hours

- [ ] Block the adversary IPv4 hosts in Appendix A, and alert on any internal host that contacts them.
- [ ] Hunt for the 15 file hashes, the script names `find_secret*.js`, `evil-ldap.py`, `evil-mysql-blob.py`, `redir_server.py` and `redir2.py`, and the JWKS key ID `hktest1`.
- [ ] Rotate any API key that ever sat in an exposed `.env` file, agent profile or session log.

Within 72 hours (application and platform teams)

- [ ] **Desktop apps.** Treat Electron `.asar` bundles and bundled `.cjs` CLI scripts as readable source. Replace static `OAUTH_APP_SECRET` values with Authorization Code plus PKCE (`S256`). Enable the `@electron/fuses` setting `EnableEmbeddedAsarIntegrityValidation`.
- [ ] **JWT verification.** Pin trusted JWKS origins. Reject unpinned `jku` headers and unknown `kid` values such as `hktest1`.
- [ ] **Redirects.** Re-check `169.254.0.0/16`, `100.100.100.200/32` and `127.0.0.0/8` on every 301, 302, 303, 307 and 308 hop.

Within 7 to 30 days

- [ ] **Cloud metadata.** On AWS, require IMDSv2 (`HttpTokens=required`). On Alibaba Cloud ECS, use hardened mode (`HttpTokens=required`, `HttpEndpoint=enabled`). On Azure, enforce the `Metadata: true` header and host firewall rules. On DigitalOcean Droplets, block non-root egress to `169.254.169.254`.
- [ ] **Java tiers.** Block outbound traffic from application servers to untrusted LDAP (1389, 389) and MySQL (3306, 3312, 3314) ports. Remove `autoDeserialize` and `queryInterceptors` from Connector/J connection strings, or upgrade to Connector/J 8.2.0 or later, where `autoDeserialize` is removed.
- [ ] **H2.** Disable H2 web consoles for remote users and forbid `INIT=RUNSCRIPT` in JDBC URLs.

- [ ] **Agent dotfiles.** Require an explicit opt-in before an agent framework reads `~/.claude/.credentials.json` or the Keychain. Look for ad-hoc HTTP servers (`python3 -m http.server`) inside `~/.hermes`, `~/.claude` and `~/.codex`, and set those files to mode 0600.
- [ ] **Threat feeds.** Before a multi-host match goes on a block list, check what it is. Wordlist entries, documentation URLs, cloud metadata addresses, loopback URLs and bare shell prefixes are not shared infrastructure. Keep a full reverse shell with a routable IP and port as a real finding.

CHAPTER 08

Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

How the Suricata signatures work 27

How the other rules work 27

08 Detection Rules

The full rules are in Appendix B. Each file targets one stage of the activity in this report.

Rule file	What it catches
`suricata_ide_secrets_jndi_and_jwks_ssrf.rules`	Suricata 7: seven signatures for the rogue JWKS response, metadata redirects, the H2 payload, Hermes path probes, raw-IP JWKS fetches, the reverse-shell script and rogue LDAP responses
`sentinel_electron_asar_and_imds_jwks_hunt.kql`	Microsoft Sentinel and Defender XDR: ASAR scraping commands, extractor or credential file writes, and contact with staging hosts or rogue Java ports
`sigma_electron_asar_unpack_and_cli_cred_theft.yml`	Sigma (3 rules): ASAR secret scraping, non-official access to agent credential stores, and JVM reverse shells
`yara_electron_asar_jndi_blob_and_jwks_toolkits.yar`	YARA (4 rules): the `find_secret` extractors, the LDAP and MySQL servers, the H2 payload, and the JWKS and redirector kit

How the Suricata signatures work

Signatures 9902601 to 9902606 read HTTP buffers. They need cleartext HTTP (ports 8080 to 8085 in this data) or TLS-inspected proxy or egress traffic. Signature 9902607 reads raw TCP on port 1389.

SID	Looks for	Why
9902601	Inbound HTTP 200 response whose body holds `"keys"` and `"hktest1"`	Catches the rogue JWKS keyset used for `jku` abuse (RFC 7515)
9902602	HTTP 301, 302, 303, 307 or 308 whose `Location` points to an AWS, Alibaba, Azure, DigitalOcean or GCP metadata path, including encoded forms of `169.254.169.254`	Catches redirect-based SSRF into cloud metadata
9902603	Inbound HTTP 200 body with `CREATE ALIAS` and `Runtime.getRuntime().exec`	Catches the H2 RCE script in transit (T1190)
9902604	HTTP request for `/.hermes/.env`, a profile `.env`, `.anthropic_oauth.json` or `sessions/`	Catches probes for exposed agent credential stores
9902605	Outbound request for `/jwks.json` to a raw IPv4 host	Catches a verifier fetching keys from an unpinned, IP-only origin
9902606	Outbound request for `/hr.sh`	Catches a fetch of the staged reverse-shell script (T1059.004)
9902607	`javaSerializedData` in traffic from external TCP port 1389	Catches a rogue JNDI LDAP answer reaching an internal Java host

How the other rules work

The KQL hunt has a 14-day lookback and three branches. The first reads process events. It flags `asar extract`, `find_secret`, or commands that touch `app.asar` or `cli-bundle.cjs` together with `OAUTH`, `SECRET`, `readFileSync` or `hostProcessLocalEnv`. It also flags non-official processes that reference agent credential files.

The second branch reads file events. It flags new `find_secret` files and writes to `.credentials.json`, `.anthropic_oauth.json` or `.env` inside `.claude` or `.hermes` by processes other than the official CLI. This matters because Defender file events record writes, not plain reads.

The third branch reads network events. It flags contact with the 12 staging hosts and Java processes connecting to public ports 1389, 3312, 3314 or 4444. It also flags plain-HTTP JWKS fetches from raw IP addresses.

The three Sigma rules cover Linux, macOS and Windows process creation:

Sigma rule	Fires when	Main false positive
ASAR secret scraping	`asar extract`, or a command touching `app.asar` or `cli-bundle.cjs` with `OAUTH`, `APP_SECRET`, `readFileSync` or `hostProcessLocalEnv`, or `node` running `find_secret`	Authorized app security audits and packaging pipelines
Agent credential store access	A non-official process references `~/.claude/.credentials.json` or `~/.hermes/` tokens, queries the Keychain for Claude entries, or runs `python -m http.server` inside `.hermes`, `.claude` or `.codex`	Developer debugging
JVM reverse shell	A Java process spawns `sh`, `bash`, `cmd.exe` or PowerShell with `/dev/tcp/`, `nc -e`, `ncat`, `/hr.sh`, `/evil1.sql` or `:8081/r?d=` in the command line	Security test harnesses; `gradle`, `maven`, `jenkins`, `surefire` and `bazel` builds are excluded

Each YARA rule needs two or more independent strings and a small file size:

YARA rule	Fires on
`ODIN_Electron_IDE_ASAR_Secret_Extractor`	Files under 50 KB that combine `readFileSync`, an `app.asar` path and a secret or closure name such as `OAUTH_APP_SECRET` or `hostProcessLocalEnv`
`ODIN_Rogue_JNDI_LDAP_and_MySQL_JDBC_Blob_Kit`	`javaSerializedData` with the BER encoder, or the MySQL packet helper with `autoDeserialize` or the `5.7.36-evilblob` / `5.7.36-debug` banners
`ODIN_H2_Database_Create_Alias_Java_RCE`	Scripts with `CREATE ALIAS IF NOT EXISTS`, `Runtime.getRuntime().exec` and `java[.]net.URLEncoder.encode` together
`ODIN_Rogue_JWKS_and_Cloud_IMDS_SSRF_Redirector`	A keyset with `"keys"`, `"hktest1"` and `"RS256"`, or a `BaseHTTPRequestHandler` script that sets `Location` to a metadata address or a redirect path such as `/r-ram`

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Adversary hosts 30
File hashes 30
MITRE ATT&CK mapping 31

09 Indicators

The full list is in Appendix A. The published set has 27 indicators; the dual-use Host W address is withheld.

Type	Count
Adversary IPv4 hosts	12
SHA-256 file hashes of staged tools	15
Exposed developer gateways or unattributed exposure hosts	0 published

Adversary hosts

Host	Port	Role
202.162.109[.]5	8081	Java JNDI LDAP, MySQL BLOB and H2 exploit workbench
143.92.57[.]247	4444	Reverse-shell, H2 callback and script callback listener
129.226.71[.]134	8899	Cloud, API and DNS recon workbench
38.246.253[.]11	8085	Rogue RS256 JWKS and metadata redirector host
43.230.171[.]150	8080	AD and SQL dropper and credential staging host
38.207.179[.]179	8088	C2 payload and command staging server
51.194.215[.]35	8080	Shell-dropper staging host
8.137.114[.]242	8080	GoPhish and recon tooling workbench
144.172.99[.]199	8080	OAST and DNSLog callback harvest host
8.138.12[.]59	9999	Fastjson exploitation and token harvest workbench
38.60.227[.]210	8082	GitHub-based C2 agent and cryptominer dropper host
47.109.23[.]77	8888	JNDI class-loader and botnet staging host

File hashes

File	Size	Host	SHA-256
`/find_secret.js`	957 B	Host W	`58d60d46032f0773450f3423c590e33def5c8d82d99ea7d3eeb030e48bcf75da`
`/find_secret10.js`	1,015 B	Host W	`fa4c8b7e32a988708b8be3f242f8da7b4950317e744d25dc29562f2d7668b89b`
`/find_secret15.js`	936 B	Host W	`ae94f76f756ba03d649fd1cfd5d5e4499964d35d240004ae1201f2d69a6ff561`
`/find_secret19.js`	706 B	Host W	`82bf3e60b1a88f3aec71340135071c7ef274bbc79d0af957feda96ef03ecd722`
`/find_secret21.js`	1,148 B	Host W	`67e0a6cd4f9d33100b5586db7fc520a9c3bdf552efba25eaa26193a271cacb96`
`/evil-ldap.py`	3,698 B	202.162.109[.]5	`f168a63ff299f329042018ac619a5798ff7f32db84e00d695486dbdb52e230d5`
`/evil-mysql-blob.py`	3,326 B	202.162.109[.]5	`fd02682812241e87923e1dd84c143172311e2fd2ccd96e24102902712d94c3bf`
`/S.py`	3,630 B	202.162.109[.]5	`2905df5e743e78e506301a5eb5db64d5ee2693ac1b2db4f471e3752f27814240`

File	Size	Host	SHA-256
`/evil1.sql`	505 B	202.162.109[.]5	`1c17f27b9f5dac7c800e1e8525d51eaa776de908f55f17acd2acc8d9f465fd0f`
`/hr.sh`	56 B	202.162.109[.]5	`6d52a91e2c376bb58f95d67b34bbdf6c79215d80074d77f19f21c551c56cf8fb`
`/redir_server.py`	897 B	38.246.253[.]11	`a6d229a047b79585c56440d4cf51bffda83431f4917d592abc15ca50bff5093a`
`/redir2.py`	826 B	38.246.253[.]11	`ecd9a613e61a457652b9a587e2c698687056ae961d743cfeb0d80235d69af23b`
`/jwks.json`	438 B	38.246.253[.]11	`ddf4f0ffd2f5a3d1f1e68801975a1788de48079f4746ff7f41eb6c14218796b7`
`/apienum.py`	1,925 B	129.226.71[.]134	`8215966f15ad99df70d12359b5357322fb2e4d6a9b47b0e97f47cbc25b823b68`
`/dnsbrute.py`	1,234 B	129.226.71[.]134	`5ef1c728993da674a1ce87e7dcce45fb377ef905338a6e0c553cb11b6df7ea59`

Supporting hunting values: JWKS key ID `hktest1`, MySQL banners `5.7.36-evilblob` and `5.7.36-debug`, the log line `evil-ldap on :1389`, and H2 alias `HR_X`.

Withheld: the two developer gateways, the four developer or unattributed exposure hosts, and the IDE vendor and install paths. Also withheld are the staged private keys, the wallet keystore, the redirector's self-probe address, and all exposed API keys and tokens.

MITRE ATT&CK mapping

Tactic	Technique	Where it appears	Detection
Credential Access	T1552.001 Credentials In Files	`find_secret*.js` scraping `app.asar`; exposed `.env` files	Sigma rule 1; YARA extractor rule; KQL branches 1 and 2
Credential Access	T1528 Steal Application Access Token	Recovering an embedded OAuth client secret; borrowing CLI OAuth tokens	Sigma rules 1 and 2; SID 9902604
Credential Access	T1555.001 Keychain	Hermes fallback code that queries the macOS Keychain	Sigma rule 2
Initial Access	T1190 Exploit Public-Facing Application	Rogue JNDI LDAP, MySQL BLOB and H2 servers on 202.162.109[.]5	SID 9902603, 9902607; YARA JNDI and H2 rules
Credential Access	T1552.005 Cloud Instance Metadata API	Redirectors to `100.100.100.200` RAM credentials	SID 9902602; YARA JWKS and redirector rule
Execution	T1059.004 Unix Shell	`/hr.sh` reverse shell to 143.92.57[.]247:4444	SID 9902606; Sigma rule 3

10

CHAPTER 10

Scope and Limits

IN SHORT

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

10 Scope and Limits

- **Time frame.** Findings use an evidence cutoff of 15 November 2026. The 25 October 2026 date in Chapter 4 is a timestamp inside the attacker's own log, not a start date for our observation.
- **Staged does not mean used.** The scripts, kits and keys show intent and capability. We did not see any of them succeed against a victim. No IDE secret, borrowed CLI token or accepted forged JWT was found.
- **Presence is not validity.** A key or C2 URL found on a server may be expired or fake. We did not test any key.
- **Dual use.** Account names on Host W leave open that it is an exposed red-team server. Authorization is unknown, so its address is withheld; its script hashes stay published as tool indicators.
- **Developer hosts are not attackers.** The Hermes gateways and other exposure hosts look like misconfigured developer machines. They are withheld and not on any block list.
- **No hidden network.** Apart from 7 values shared by 43.230.171[.]150, nothing in this data links the adversary hosts to each other.
- **Rule testing.** Every published rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma and Microsoft's Kusto.Language analyzer, checked on 20 November 2026). The rules were not replayed against live traffic, so test and tune them in your own environment before production use.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. Electron. ASAR Archives. https://www.electronjs.org/docs/latest/tutorial/asar-archives (verified 15 November 2026)
2. IETF. RFC 8252, OAuth 2.0 for Native Apps (BCP 212). https://datatracker.ietf.org/doc/html/rfc8252 (verified 15 November 2026)
3. MITRE ATT&CK. T1190 Exploit Public-Facing Application. https://attack.mitre.org/techniques/T1190/ (verified 15 November 2026)
4. IETF. RFC 7515, JSON Web Signature, Section 4.1.2 (`jku`), and RFC 7517, JSON Web Key. https://datatracker.ietf.org/doc/html/rfc7515 (verified 15 November 2026)
5. MITRE ATT&CK. T1552.005 Cloud Instance Metadata API. https://attack.mitre.org/techniques/T1552/005/ (verified 15 November 2026)
6. MITRE ATT&CK. T1552.001 Credentials In Files. https://attack.mitre.org/techniques/T1552/001/ (verified 15 November 2026)
7. Alibaba Cloud. ECS: Access instance metadata using hardened-only mode. https://help.aliyun.com/en/ecs/user-guide/access-instance-metadata-using-hardened-only-mode (verified 15 November 2026)
8. Oracle. MySQL Connector/J 8.2.0 release notes. https://dev.mysql.com/doc/relnotes/connector-j/en/news-8-2-0.html (verified 15 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 (27) 37

B. Detection rules (4 files) 39

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
R09-IOC-01	IPv4	202.162.109[.]5	8081	Java JNDI LDAP, MySQL JDBC BLOB & H2 RCE Exploit Workbench	BLOCK
R09-IOC-02	IPv4	143.92.57[.]247	4444	Downstream Reverse-Shell, H2 Exfiltration & XSS Listener	BLOCK
R09-IOC-03	IPv4	129.226.71[.]134	8899	Cloud/API/DNS Reconnaissance & Injection Validation Workbench	BLOCK
R09-IOC-04	IPv4	38.246.253[.]11	8085	Rogue RS256 JWKS (RFC 7515 jku) & HTTP 301/302/307/308 Redirector	BLOCK
R09-IOC-05	SHA-256	67e0a6cd4f9d33100b5586db7fc520a9c3bdf552efba25eaa26193a271cacb96	8000	Electron AI IDE ASAR Nearest-Neighbor Closure Locator #21 SHA-256	HUNT
R09-IOC-06	SHA-256	58d60d46032f0773450f3423c590e33def5c8d82d99ea7d3eeb030e48bcf75da	8000	Electron AI IDE app.asar & CLI Bundle OAuth Secret Scanner #1 SHA-256	HUNT
R09-IOC-07	SHA-256	fa4c8b7e32a988708b8be3f242f8da7b4950317e744d25dc29562f2d7668b89b	8000	Electron AI IDE hostProcessLocalEnv Spawn-Host Inspector #10 SHA-256	HUNT
R09-IOC-08	SHA-256	ae94f76f756ba03d649fd1cfd5d5e4499964d35d240004ae1201f2d69a6ff561	8000	Electron AI IDE ASAR Module Scope Parser #15 SHA-256	HUNT
R09-IOC-09	SHA-256	82bf3e60b1a88f3aec71340135071c7ef274bbc79d0af957feda96ef03ecd722	8000	Electron AI IDE ASAR Minified Closure Locator #19 SHA-256	HUNT
R09-IOC-10	SHA-256	2905df5e743e78e506301a5eb5db64d5ee2693ac1b2db4f471e3752f27814240	3314	Configurable Rogue MySQL Protocol BLOB Server (TCP 3314) SHA-256	HUNT
R09-IOC-11	SHA-256	f168a63ff299f329042018ac619a5798ff7f32db84e00d695486dbdb52e230d5	1389	Rogue Log4Shell javaSerializedData LDAP Server (TCP 1389) SHA-256	HUNT
R09-IOC-12	SHA-256	fd02682812241e87923e1dd84c143172311e2fd2ccd96e24102902712d94c3bf	3312	Rogue MySQL JDBC autoDeserialize BLOB Server (TCP 3312) SHA-256	HUNT
R09-IOC-13	SHA-256	1c17f27b9f5dac7c800e1e8525d51eaa776de908f55f17acd2acc8d9f465fd0f	8081	H2 Database CREATE ALIAS Java Runtime RCE Payload SHA-256	HUNT
R09-IOC-14	SHA-256	6d52a91e2c376bb58f95d67b34bbdf6c79215d80074d77f19f21c551c56cf8fb	8081	Bash /dev/tcp/143.92.57[.]247/4444 Reverse Shell Script SHA-256	HUNT
R09-IOC-15	SHA-256	a6d229a047b79585c56440d4cf51bffda83431f4917d592abc15ca50bff5093a	8085	HTTP 302 IMDS (100.100.100.200 RAM) SSRF Redirector SHA-256	HUNT
R09-IOC-16	SHA-256	ecd9a613e61a457652b9a587e2c698687056ae961d743cfeb0d80235d69af23b	8085	Multi-Code (301/302/307/308) HTTP Redirect Bypass Server SHA-256	HUNT
R09-IOC-17	SHA-256	ddf4f0ffd2f5a3d1f1e68801975a1788de48079f4746ff7f41eb6c14218796b7	8085	Rogue RS256 JWKS Public Key Set (kid=hktest1) SHA-256	HUNT
R09-IOC-18	SHA-256	8215966f15ad99df70d12359b5357322fb2e4d6a9b47b0e97f47cbc25b823b68	8899	Recon Workbench API Endpoint Enumerator SHA-256	HUNT
R09-IOC-19	SHA-256	5ef1c728993da674a1ce87e7dcce45fb377ef905338a6e0c553cb11b6df7ea59	8899	Recon Workbench Subdomain Brute-Forcer SHA-256	HUNT
R09-IOC-20	IPv4	43.230.171[.]150	8080	Multi-Host AD/SQL Dropper & Credential Staging Host	BLOCK

ID	Type	Indicator	Port	Context	Action
R09-IOC-21	IPv4	38.207.179[.]179	8088	High-Density C2 Payload & Command Staging Server	BLOCK
R09-IOC-22	IPv4	51.194.215[.]35	8080	Automated Shell-Dropper Staging Infrastructure	BLOCK
R09-IOC-23	IPv4	8.137.114[.]242	8080	Exposed Gophish & Reconnaissance Tooling Workbench	BLOCK
R09-IOC-24	IPv4	144.172.99[.]199	8080	OAST & DNSLog Out-of-Band Callback Harvest Host	BLOCK
R09-IOC-25	IPv4	8.138.12[.]59	9999	Fastjson Exploitation & Session Token Harvest Workbench	BLOCK
R09-IOC-26	IPv4	38.60.227[.]210	8082	GitHub-C2 Agent & Cryptominer Dropper Staging Host	BLOCK
R09-IOC-27	IPv4	47.109.23[.]77	8888	JNDI Exploit Class-Loader & Botnet Staging Host	BLOCK

B Detection Rules

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

sentinel_electron_asar_and_imds_jwks_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Report #09 Microsoft Sentinel & Defender XDR Hunt (TLP:CLEAR)
// Scope: Electron ASAR Secret Extraction, CLI OAuth Dotfile Staging, and Java/JWKS Callbacks
let Lookback = 14d;
let OffensiveStagingHosts = dynamic(["202.162.109.5", "143.92.57.247", "129.226.71.134", "38.246.253.11",
    "43.230.171.150", "38.207.179.179", "51.194.215.35", "8.137.114.242", "144.172.99.199",
    "8.138.12.59", "38.60.227.210", "47.109.23.77"]);
let RogueJavaPorts = dynamic([1389, 3312, 3314, 4444]);
union (
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where ProcessCommandLine has "asar extract" or ProcessCommandLine contains "find_secret"
        or (ProcessCommandLine has_any ("app.asar", "cli-bundle.cjs") and (ProcessCommandLine contains "OAUTH"
or ProcessCommandLine contains "SECRET" or ProcessCommandLine has_any ("readFileSync", "hostProcessLocalEnv")))
        or (ProcessCommandLine has_any (".claude/.credentials.json", ".anthropic_oauth.json", ".hermes/.env"))
    | project Timestamp, DeviceName, SignalType = "Endpoint_CLI_ASAR_or_OAuth_Command", ActorProcess = FileName,
Detail = ProcessCommandLine
), (
    DeviceFileEvents
    | where Timestamp >= ago(Lookback) and ActionType in ("FileCreated", "FileModified", "FileRenamed")
    | where FileName contains "find_secret"
        or (FolderPath has_any (".claude", ".hermes") and FileName in~ (".credentials.json",
".anthropic_oauth.json", ".env"))
    | project Timestamp, DeviceName, SignalType = "Endpoint_Extractor_Drop_or_OAuth_FileWrite", ActorProcess =
InitiatingProcessFileName, Detail = strcat(FolderPath, "/", FileName)
), (
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback) and ActionType in ("ConnectionSuccess", "HttpConnectionInspected")
    | extend InspectedFields = parse_json(AdditionalFields)
    | extend HttpUri = tostring(InspectedFields.uri), HttpHost = tostring(InspectedFields.host)
    | where RemoteIP in (OffensiveStagingHosts)
        or (InitiatingProcessFileName in~ ("java", "java.exe") and RemoteIPType == "Public" and RemotePort in
(RogueJavaPorts))
        or (RemoteIPType == "Public" and ((RemoteUrl has "/jwks.json" and RemoteUrl matches regex @"^http://
\d{1,3}(\.\d{1,3}){3}(:\d+)?/.*jwks\.json")
            or (HttpUri endswith "/jwks.json" and (HttpHost matches regex @"^\d{1,3}(\.\d{1,3}){3}(:\d+)?$" or
isempty(HttpHost)))))
    | project Timestamp, DeviceName, SignalType = "Network_StagingHost_JNDI_MySQL_or_HTTP_JWKS", ActorProcess =
InitiatingProcessFileName, Detail = strcat(RemoteIP, ":", RemotePort, " ", iff(isnotempty(HttpUri), HttpUri,
RemoteUrl))
)
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), SampleDetails =
make_set(Detail, 5) by SignalType, DeviceName, ActorProcess
| order by EventCount desc
```

sigma_electron_asar_unpack_and_cli_cred_theft.yml

SIGMA

```
title: Electron ASAR Archive Secret Scraping via Node or Asar CLI
id: 99a41e12-5c09-4e11-8d21-090909000001
status: experimental
description: Detects cross-platform (Linux, macOS, Windows) ad-hoc Node.js scripts or asar CLI invocations
unpacking or slicing Electron app.asar and companion CLI CJS bundles via command-line arguments to extract
embedded OAuth client secrets.
author: Odin by Brandefense
date: 2026-11-16
tags:
  - attack.credential_access
```

```
    - attack.t1552.001
  logsource:
    category: process_creation
  detection:
    selection_asar_cli:
      CommandLine|contains:
        - 'asar extract'
        - 'asar e '
    selection_inline_bundle:
      CommandLine|contains:
        - 'resources/app.asar'
        - 'resources\app.asar'
        - 'Contents/Resources/app.asar'
        - 'app.asar.unpacked'
        - 'cli-bundle.cjs'
    selection_inline_intent:
      CommandLine|contains:
        - 'OAUTH'
        - 'APP_SECRET'
        - 'readFileSync'
        - 'hostProcessLocalEnv'
    selection_extractor_script:
      Image|endswith:
        - '/node'
        - '\node.exe'
      CommandLine|contains: 'find_secret'
    condition: selection_asar_cli or (selection_inline_bundle and selection_inline_intent) or
  selection_extractor_script
  falsepositives:
    - Authorized desktop application security audits and CI/CD packaging pipelines
  level: high
  ---
  title: Unauthorized CLI Access or Ad-Hoc HTTP Exposure of Claude and Hermes Credential Stores
  id: 99a41e12-5c09-4e11-8d21-090909000002
  status: experimental
  description: Detects non-official CLI processes referencing ~/.claude/.credentials.json or ~/.hermes/ token
  stores in command lines, macOS Keychain queries for Claude OAuth entries, or ad-hoc Python HTTP servers launched
  inside AI agent dotfile directories.
  author: Odin by Brandefense
  date: 2026-11-16
  tags:
    - attack.credential_access
    - attack.t1528
    - attack.t1555.001
  logsource:
    category: process_creation
  detection:
    selection_cred_paths:
      CommandLine|contains:
        - '.claude/.credentials.json'
        - '.claude\.credentials.json'
        - '.hermes/.anthropic_oauth.json'
        - '.hermes\.anthropic_oauth.json'
        - '.hermes/.env'
        - '.hermes\.env'
    selection_macos_keychain:
      Image|endswith: '/security'
      CommandLine|contains:
        - 'find-generic-password'
        - 'dump-keychain'
    selection_macos_target:
      CommandLine|contains:
        - 'Claude'
        - 'anthropic'
    selection_adhoc_http_proc:
      Image|endswith:
        - '/python'
        - '/python3'
        - '\python.exe'
      CommandLine|contains:
        - 'http.server'
        - 'SimpleHTTPServer'
```

```
    selection_adhoc_http_cwd:
      CurrentDirectory|contains:
        - '/.hermes'
        - '\.hermes'
        - '/.claude'
        - '\.claude'
        - '/.codex'
        - '\.codex'
    selection_adhoc_http_arg:
      CommandLine|contains:
        - '.hermes'
        - '.claude'
        - '.codex'
    condition: selection_cred_paths or (selection_macos_keychain and selection_macos_target) or
  (selection_adhoc_http_proc and (selection_adhoc_http_cwd or selection_adhoc_http_arg))
  falsepositives:
    - Developer backup scripts explicitly archiving dotfile configurations
    - The official vendor CLI reading its own credential file; tune by exact signed executable path in the SIEM
  level: high
  ---
  title: Java Process Spawning Reverse-Shell or Exploit Callback Subshell
  id: 99a41e12-5c09-4e11-8d21-090909000003
  status: experimental
  description: Detects JVM application processes spawning shells with explicit reverse-shell redirection or
  exploit-callback primitives following H2 CREATE ALIAS, Log4Shell JNDI, or MySQL JDBC autoDeserialize exploitation
  across Linux, macOS, and Windows.
  author: Odin by Brandefense
  date: 2026-11-16
  tags:
    - attack.initial_access
    - attack.t1190
    - attack.execution
    - attack.t1059.004
  logsource:
    category: process_creation
  detection:
    selection_parent:
      ParentImage|endswith:
        - '/java'
        - '\java.exe'
    selection_shell:
      Image|endswith:
        - '/sh'
        - '/bash'
        - '/dash'
        - '/zsh'
        - '\cmd.exe'
        - '\powershell.exe'
        - '\pwsh.exe'
    selection_payload:
      CommandLine|contains:
        - '/dev/tcp/'
        - 'nc -e'
        - 'nc -c'
        - 'ncat '
        - '/hr.sh'
        - '/evil1.sql'
        - ':8081/r?d='
    condition: selection_parent and selection_shell and selection_payload
  falsepositives:
    - Custom security-validation harnesses executing simulated reverse-shell commands inside test JVMs
  level: high
```

suricata_ide_secrets_jndi_and_jwks_ssrf.rules

SURICATA

```
# Odin by Brandefense - Report #09 Suricata 7.x Detection Suite
# Classification: TLP:CLEAR | Scope: Electron IDE Exposure, Java JNDI/JDBC Workbenches & Rogue JWKS/IMDS
Redirectors
# Deployment Note: Rules 9902601-9902606 inspect HTTP sticky buffers over cleartext HTTP or TLS-inspected
reverse-proxy/egress traffic; pair with endpoint telemetry (KQL/Sigma) for uninspected HTTPS flows.
# 1. Rogue RS256 JWKS key-injection response serving attacker key ID (38.246.253.11:8085/jwks.json, RFC 7515
jku / RFC 7517)
alert http !$HOME_NET any -> $HOME_NET any (msg:"ODIN-IDE Rogue RS256 JWKS Public Key Set Response
(kid=hktest1)"; flow:established,to_client; http.stat_code; content:"200"; http.response_body;
content:"\"keys\""; content:"\"hktest1\""; fast_pattern; distance:0; classtype:web-application-attack; sid:
9902601; rev:4;)
# 2. HTTP 301/302/303/307/308 SSRF redirect pivoting to Alibaba ECS, AWS, Azure, DigitalOcean, or GCP metadata
endpoints (redir_server.py & redir2.py)
alert http !$HOME_NET any -> $HOME_NET any (msg:"ODIN-IDE Cloud IMDS Metadata SSRF HTTP Redirect Bypass";
flow:established,to_client; http.stat_code; pcre:"/^30[12378]$/"; http.location; content:"http://"; nocase;
startswith; fast_pattern; pcre:"/^https?:\/\/(100\.100\.100\.200|169\.254\.169\.254|\[?fd00:ec2::254\]?|
metadata\.google\.internal|2852039166|0xa9fea9fe)(:\d+)?\/(latest\/(meta-data|user-data)|metadata\/(instance|
v1(\.json)?)|computeMetadata)(\/|$|\?)/i"; classtype:attempted-admin; sid:9902602; rev:4;)
# 3. Inbound H2 Database JDBC RUNSCRIPT CREATE ALIAS inline Java Runtime.exec payload (GET /evil1.sql 200 OK)
alert http !$HOME_NET any -> $HOME_NET any (msg:"ODIN-IDE Inbound H2 JDBC RUNSCRIPT CREATE ALIAS Java RCE
Payload"; flow:established,to_client; http.stat_code; content:"200"; http.response_body; content:"CREATE ALIAS";
nocase; content:"Runtime.getRuntime().exec"; fast_pattern; distance:0; classtype:web-application-attack; sid:
9902603; rev:4;)
# 4. Unauthenticated HTTP request probing Hermes AI agent .env, OAuth, or session store paths (/.hermes/)
alert http any any -> $HOME_NET any (msg:"ODIN-IDE HTTP Probe for Hermes CLI Agent Credential or Session Store
Path"; flow:established,to_server; http.uri; content:"/.hermes/"; nocase; fast_pattern; pcre:"/\/\.hermes\/
(\.env|profiles\/[^\/]+\/\.env|\.anthropic_oauth\.json|sessions\/)/i"; classtype:credential-theft; sid:9902604;
rev:4;)
# 5. Outbound RFC 7515 jku verification fetch to a raw IPv4-hosted JWKS endpoint (/jwks.json or /.well-known/
jwks.json)
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-IDE Outbound JWS jku Fetch to Raw IPv4 JWKS Endpoint (/
jwks.json)"; flow:established,to_server; http.uri; content:"/jwks.json"; nocase; fast_pattern; http.host; pcre:"/
^\d{1,3}(\.\d{1,3}){3}(:\d+)?$/"; classtype:attempted-admin; sid:9902605; rev:4;)
# 6. Outbound fetch to staged Bash reverse-shell dropper script (/hr.sh on 202.162.109.5:8081)
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-IDE Outbound Fetch to Staged Bash Reverse-Shell Dropper (/
hr.sh)"; flow:established,to_server; http.uri; content:"/hr.sh"; fast_pattern; pcre:"/\/hr\.sh(\?|$)/";
classtype:trojan-activity; sid:9902606; rev:4;)
# 7. Inbound rogue Java JNDI LDAP javaSerializedData response from external TCP port 1389 (evil-ldap.py)
alert tcp !$HOME_NET 1389 -> $HOME_NET any (msg:"ODIN-IDE Inbound Rogue JNDI LDAP javaSerializedData Response
from Port 1389"; flow:established,from_server; content:"javaSerializedData"; nocase; fast_pattern;
classtype:attempted-admin; sid:9902607; rev:4;)
```

yara_electron_asar_jndi_blob_and_jwks_toolkits.yar

YARA

```
/*
   Odin by Brandefense - Report #09 YARA Workbench & Exploitation Toolkit Ruleset
   Classification: TLP:CLEAR
*/

rule ODIN_Electron_IDE_ASAR_Secret_Extractor {
    meta:
        author = "Odin by Brandefense"
        description = "Detects Node.js scripts scraping cross-platform Electron app.asar and companion CLI CJS
bundles for embedded OAuth secrets and spawn-host closures"
        reference = "Report #09 - Host W (address withheld)"
        tlp = "CLEAR"
    strings:
        $b1 = "resources/app.asar" ascii nocase
        $b2 = "resources\\app.asar" ascii nocase
        $b3 = "Contents/Resources/app.asar" ascii nocase
        $io = "readFileSync" ascii
        $s1 = "OAUTH_APP_SECRET" ascii nocase
        $s2 = "OAUTH_CLIENT_SECRET" ascii nocase
        $s3 = "appSecret" ascii
```

```
        $s4 = "hostProcessLocalEnv" ascii
        $s5 = "spawnHostProcess" ascii
        $s6 = "tokenByAuthCode" ascii
    condition:
        filesize < 50KB and $io and (1 of ($b*)) and (1 of ($s*))
}

rule ODIN_Rogue_JNDI_LDAP_and_MySQL_JDBC_Blob_Kit {
    meta:
        author = "Odin by Brandefense"
        description = "Detects Python rogue Log4Shell BER LDAP and MySQL protocol BLOB deserialization servers"
        reference = "Report #09 - 202.162.109.5:8081"
        tlp = "CLEAR"
    strings:
        $ldap1 = "javaSerializedData" ascii
        $ldap2 = "ber_tlv(tag, content)" ascii
        $mysql1 = "autoDeserialize" ascii
        $mysql2 = "pkt(seq, data)" ascii
        $mysql3 = "5.7.36-debug" ascii
        $mysql4 = "5.7.36-evilblob" ascii
    condition:
        filesize < 60KB and (($ldap1 and $ldap2) or ($mysql1 and $mysql2) or ($mysql2 and ($mysql3 or $mysql4)))
}

rule ODIN_H2_Database_Create_Alias_Java_RCE {
    meta:
        author = "Odin by Brandefense"
        description = "Detects H2 SQL injection scripts compiling inline Java Runtime.getRuntime().exec aliases"
        reference = "Report #09 - evil1.sql"
        tlp = "CLEAR"
    strings:
        $a1 = "CREATE ALIAS IF NOT EXISTS" ascii nocase
        $a2 = "Runtime.getRuntime().exec" ascii
        $a3 = "java.net.URLEncoder.encode" ascii
    condition:
        filesize < 25KB and all of ($a*)
}

rule ODIN_Rogue_JWKS_and_Cloud_IMDS_SSRF_Redirector {
    meta:
        author = "Odin by Brandefense"
        description = "Detects rogue RS256 JWKS keysets (kid=hktest1) and Python HTTP 301/302/307/308 Cloud IMDS
SSRF redirectors"
        reference = "Report #09 - 38.246.253.11:8085"
        tlp = "CLEAR"
    strings:
        $j1 = "\"keys\"" ascii
        $j2 = "\"hktest1\"" ascii
        $j3 = "\"RS256\"" ascii
        $p_base = "BaseHTTPRequestHandler" ascii
        $p_loc_sq = "send_header('Location'" ascii
        $p_loc_dq = "send_header(\"Location\"" ascii
        $r1 = "100.100.100.200" ascii
        $r2 = "169.254.169.254" ascii
        $r3 = "/r-ram" ascii
        $r4 = "/r-meta" ascii
        $r5 = "/r301" ascii
        $r6 = "/r302" ascii
        $r7 = "/r307" ascii
        $r8 = "/r308" ascii
    condition:
        filesize < 30KB and (all of ($j*) or ($p_base and (1 of ($p_loc_*)) and (1 of ($r*))))
}
```

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/electron-ide-secret-theft-and-single-host-exploit-workbenches/
Short Read · brandefense.io/odin-research/blog/attackers-are-mining-desktop-ai-tools-for-oauth-secrets/
Defense Kit · brandefense.io/odin-research/defense-kit/