ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 24 / 30 · CREDENTIAL THEFT

Session Cloners and Stealer Fleets
Replaying Stolen Browsers Past MFA

Playwright session-cloning workbenches, wallet lures, and the Aurora, StealC, RedLine, LaplasClipper, PikaBot and Ramnit infrastructure behind them

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

193
Endpoints across 11 active stealer, clipper, loader and RAT families, on 165 distinct hosts

20
Confirmed C2 endpoints on 18 IPs; 15 of them (13 IPs) are published for blocking

80
Sampled files on the OpenClaw and Playwright session-cloning workbench (port 8888)

59
Target folders (10,650 files) on the wallet and SaaS lure workbench (port 8080)

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

Playwright session-cloning workbenches, wallet lures, and the Aurora, StealC, RedLine, LaplasClipper, PikaBot and Ramnit infrastructure behind them

01	Executive Summary	04
02	How Stolen Sessions Get Replayed	07
03	Two Exposed Workbenches	10
04	Stealers and Clippers	14
05	Loaders and RAT Backdoors	18
06	Separating Signal from Noise	21
07	What Defenders Should Do	24
08	Detection Rules	26
09	Indicators	29
10	Scope and Limits	32
11	Sources	34
A	Indicators of compromise (15)	36
B	Detection rules (4 files)	37

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense mapped 193 endpoints on 165 hosts across 11 stealer, clipper, loader and RAT families. Two exposed workbenches sit beside them. One holds an OpenClaw agent set up to drive an existing Chromium profile through headless Playwright. The other stages wallet-brand recon and SaaS cookie capture. Twenty endpoints are confirmed C2 servers. Bind sessions to devices, block the 15 published C2 endpoints, and alert when non-browser processes read Chromium cookie stores.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense mapped 193 endpoints on 165 hosts across 11 stealer, clipper, loader and RAT families. Two exposed workbenches sit beside them. One holds an OpenClaw agent set up to drive an existing Chromium profile through headless Playwright. The other stages wallet-brand recon and SaaS cookie capture. Twenty endpoints are confirmed C2 servers. Bind sessions to devices, block the 15 published C2 endpoints, and alert when non-browser processes read Chromium cookie stores.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Infostealers no longer need your password. They take the browser's cookie database, and a cookie for a logged-in session is often enough to act as the user. This report looks at both halves of that problem. The first is the commodity stealer, clipper and loader servers that collect browser data. The second is two exposed workbenches that show how an automation agent can drive a logged-in browser profile.

1. **An exposed OpenClaw workbench drives a logged-in Chromium profile with headless Playwright.**
   An open directory on port 8888 of a server in China yielded 80 sampled files. They pair Playwright's headless Chromium build 1208 with an OpenClaw agent folder named `browser-existing-session`. That folder holds a Chromium `Cookies` store, `Trust Tokens`, `History` and an NSS key database, plus an agent action log.

2. **A second workbench pairs wallet-brand recon with SaaS cookie capture.**
   A server in France exposed 59 target folders and 10,650 files on port 8080. It holds dangling-subdomain recon against a hardware crypto-wallet vendor and headless Chrome 149 scripts that save cookie jars from a design SaaS platform. Both brands are withheld.

3. **Commodity stealers and loaders supply the stolen browser data.**
   Aurora and AuroraStealer account for 18 endpoints on 9 hosts, including a 3-port hub in Germany. StealC, RedLineStealer and 14 suspected LaplasClipper nodes follow. PikaBot, ChaluboRAT and Ramnit sit in the delivery tier. Of 193 endpoints, 20 are confirmed C2 servers.

Three Decisions for Leaders

1. **Bind web sessions to the device.** Adopt Device Bound Session Credentials (DBSC) or continuous access evaluation, so a cookie copied to a headless Linux server fails when it is replayed from a new network.
2. **Block confirmed C2, watch the rest.** Block the 15 published confirmed endpoints. Keep dynamic telecom addresses, suspected clusters and old Ramnit candidates on time-bound watchlists instead of permanent blocks.
3. **Close lure surfaces and guard profile stores.** Remove dangling DNS records that a wallet or login lure could sit on. Alert when any process other than the browser reads Chromium `Cookies`, `Trust Tokens` or extension storage.

The Study in Numbers

Number	What it is
193	Endpoints across 11 active stealer, clipper, loader and RAT families, on 165 distinct hosts
20	Confirmed C2 endpoints on 18 IPs; 15 of them (13 IPs) are published for blocking
80	Sampled files on the OpenClaw and Playwright session-cloning workbench (port 8888)
59	Target folders (10,650 files) on the wallet and SaaS lure workbench (port 8080)
18	AuroraStealer (13) and Aurora (5) endpoints on 9 hosts; 6 are confirmed
14	Suspected LaplasClipper wallet-swap nodes, all on one US provider
107 of 110	Ramnit endpoints held on watchlists rather than blocked

How to use this report.

Security leaders can read Chapter 1, Chapter 7 and Chapter 10. Identity and browser-security teams should read Chapters 2 and 3. SOC and detection engineers should start with Chapters 8 and 9. CTI analysts will find the stealer, clipper and loader infrastructure in Chapters 4 to 6.

02

CHAPTER 02

How Stolen Sessions Get Replayed

IN SHORT

A browser cookie for a logged-in session can stand in for a password and an MFA prompt. This chapter sets out an analytical model of how stealers, clippers, loaders and browser automation fit together.

INSIDE THIS CHAPTER

Why cookies matter more than passwords 07

A four-part model 07

Where the endpoints sit 08

02 How Stolen Sessions Get Replayed

Why cookies matter more than passwords

Defenders used to answer an infostealer infection by resetting passwords. That no longer closes the door. Modern stealers copy the whole Chromium user-data folder: `Default/Cookies`, `Default/Trust Tokens`, `Local Storage` and the `LevelDB` folders where wallet extensions keep their data.

Identity providers check the device, the TLS fingerprint and a FIDO2 prompt at login. Once a session cookie is issued, many services simply trust it. So whoever holds a fresh cookie can often act as the user without seeing an MFA prompt.

Browser automation makes that easy at scale. Microsoft Playwright's `launchPersistentContext` API starts a Chromium instance on top of an existing user-data folder. Any script or AI agent driving that browser inherits the sessions stored in it. Playwright documents this as a normal feature for testing.

A four-part model

Figure 1 shows the four parts as an analytical model. The 193 endpoints and two workbenches were observed separately. No IP appears in more than one family, and no shared host, certificate or campaign token links the stealer servers to the workbenches.



Figure 1: Analytical model of how stealer logs and session-cloning agents could sidestep MFA

Part	What it does	Families or tools	Observed endpoints
Steal	Copies Chromium cookies, tokens and extension vaults	AuroraStealer, Aurora, StealC, RedLineStealer	31 (11 confirmed)
Clone	Drives an existing browser profile with a headless agent	OpenClaw, Playwright Chromium 1208	Workbench 1 (80 files)
Lure and clip	Stages wallet and SaaS lures, swaps copied wallet addresses	LaplasClipper, `subfinder`, Playwright scripts	14 clipper nodes, Workbench 2
Deliver and persist	Drops stealers and keeps remote access	PikaBot, TitanOmega, Ramnit, ChaluboRAT, RaptorRAT, BitRAT	148 (9 confirmed)

What the model does not show matters as much. We did not see any replay against a real account, and we saw no losses on the target side. The model explains why these pieces are worth defending against together.

Where the endpoints sit

The 193 endpoints spread across 13 tracked families, 11 of them with active endpoints. Two more families, Bumblebee and RaccoonStealer, had no active endpoints in the window. Nineteen hosts expose more than one port.

Country	Endpoints	Share	Main driver
China	78	40.4%	Ramnit
United States	35	18.1%	14 LaplasClipper and 8 TitanOmega nodes
Germany	15	7.8%	RaptorRAT, AuroraStealer hub
Netherlands	10	5.2%	RedLineStealer, StealC
Russia	7		AuroraStealer
Hong Kong	7		ChaluboRAT, Ramnit
South Korea	7		Ramnit

The confirmed C2 servers look different from the bulk. They sit mostly on European and Asian VPS providers, including Hetzner Online, Netsec Limited, RootLayer Web Services, DigitalOcean, Google Cloud, Majestic Hosting and UFO Hosting.

All 20 confirmed endpoints were seen on 13 November 2026. Suspected Aurora and Ramnit staging was still being seen on 18 November 2026. Some old Ramnit records carry first-seen dates from earlier years; those are record dates, not the period of this study.

03

CHAPTER 03

Two Exposed Workbenches

IN SHORT

Two open directories show the tooling side. One holds an OpenClaw agent wired to a logged-in Chromium profile. The other holds wallet-brand recon and Playwright cookie capture against a SaaS platform.

INSIDE THIS CHAPTER

Workbench 1: OpenClaw agent on a logged-in Chromium profile 10

Workbench 2: wallet-brand recon and SaaS cookie capture 11

03 Two Exposed Workbenches

Figure 2 maps each tool group to how it is used, what it targets and which rule covers it.

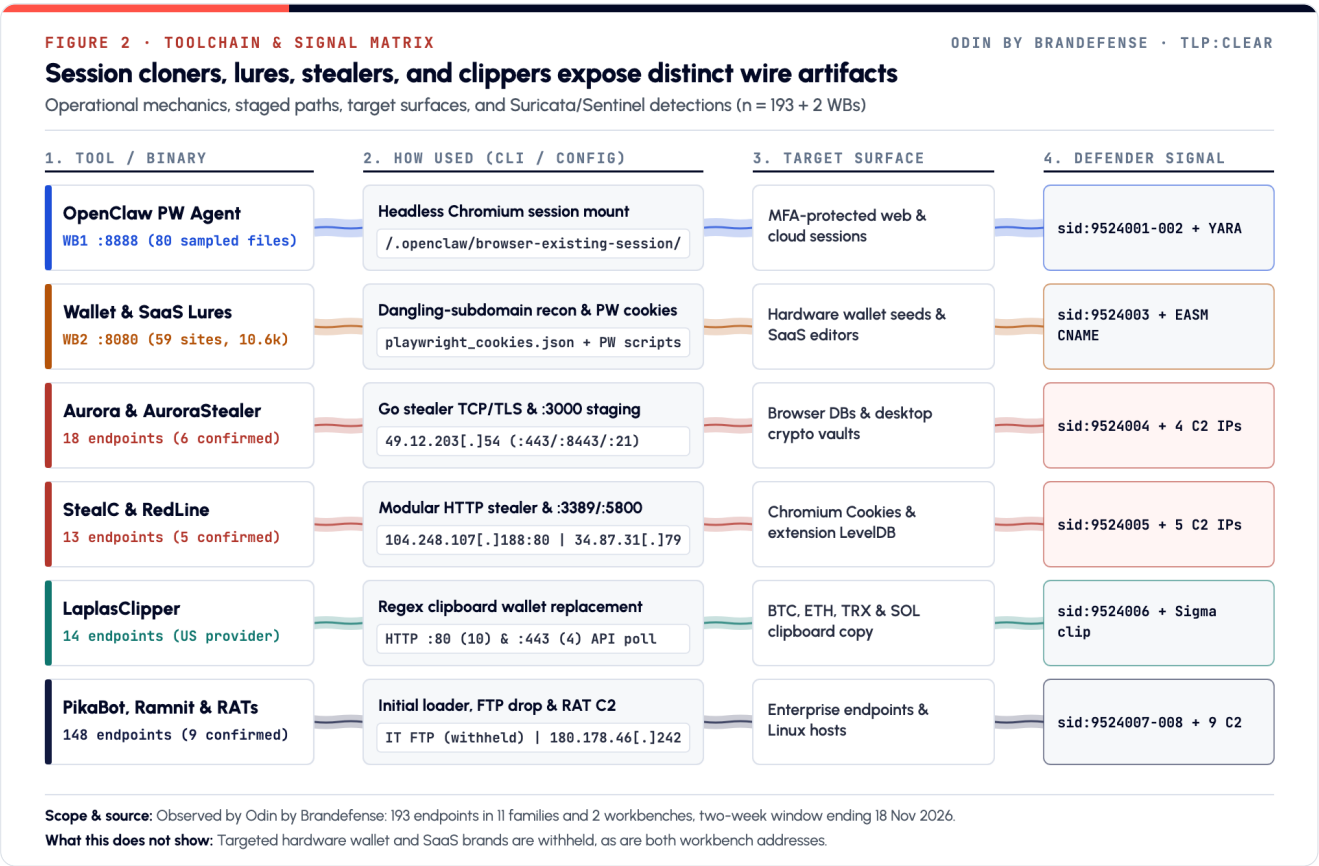


Figure 2: Session cloners, lures, stealers and clippers expose distinct wire artifacts

Workbench 1: OpenClaw agent on a logged-in Chromium profile

Workbench 1 is a cloud server in China. It exposed an open directory on port 8888, and 80 files were sampled from it. Its address is withheld, because the owner and the origin of the browser data are not established.

The files pair two things. The first is Playwright's headless Chromium build 1208, stored under `/.cache/ms-playwright/chromium_headless_shell-1208/`. The second is an OpenClaw agent folder, `/.openclaw/`, with a subfolder named `browser-existing-session`. OpenClaw is an autonomous agent that can drive a browser.

That subfolder is a full Chromium profile. It holds a `Cookies` store, `Trust Tokens`, `History`, a cached extension package and an NSS key database (`/.pki/nssdb/key4.db`). The agent logs every step it takes to a trajectory file and keeps its state in SQLite. Messaging connector plugins for QQ (`qqbot`) and DingTalk (`dingtalk-connector`) are installed beside it.

Path on Workbench 1	Size (bytes)	Role
`/.openclaw/browser-existing-session/component_crx_cache/` (one cached file)	9,541,687	Cached Chromium component or extension package inside the mounted profile
`/.skillhub/skills_index.local.json`	9,257,868	Local agent skill registry, next to `skills_store_cli.py` (156,228 bytes)
`/.cache/ms-playwright/chromium_headless_shell-1208/../libvk_swiftshader.so`	4,361,888	Software GPU renderer for headless Chromium 1208, with `libGLESv2.so`
`/.openclaw/agents/main/sessions/<session>.trajectory.jsonl`	2,355,006	Step-by-step log of the agent's browser actions
`/.openclaw/state/openclaw.sqlite`	1,245,184	Agent state database, with a 156,592-byte `.sqlite-wal` journal
`/.openclaw/browser-existing-session/Default/History`	163,840	Chromium navigation history
`/.openclaw/browser-existing-session/Default/Trust Tokens`	36,864	Chromium Trust Tokens store; `key4.db` is the same size
`/.openclaw/browser-existing-session/Default/Cookies`	20,480	Chromium cookie store the headless browser loads

The 80 sampled files break down as 22 `.json`, 14 extensionless Chromium SQLite and cache files, 7 `.xlsx`, 6 `.so`, 5 `.md`, 3 `.jsonl`, 3 `.db` and 2 `.sqlite`, plus a few singletons.

> **What this does and does not show.**
>
> The files show the mechanism: an agent set up to drive a logged-in browser profile without a fresh login. They do not show where the profile came from. It could be stolen stealer output, or the owner's own browser copied for automation. This report publishes no cookie values and no file hashes from this server.

Workbench 2: wallet-brand recon and SaaS cookie capture

Workbench 2 is a server in France with an open directory on port 8080. It holds 59 target folders and 10,650 files. Its address is withheld. An unhardened offensive-security contractor box is a possible explanation, so we treat its owner as unknown.

Among the 419 sampled entries, the largest groups are 131 `.jpeg` screenshots, 90 `.txt` recon lists and 67 `.md` reports. Next come 33 `.html` captures, 31 `.json` API and cookie logs, 31 `.png` captures and 20 `.js` scripts. Other file types make up the rest.

Two activities stand out:

- **Wallet-brand recon.** One folder targets a hardware crypto-wallet vendor. It looks for dangling staging subdomains, the kind of abandoned `CNAME` record on which an attacker could host a seed-phrase recovery or login lure under the vendor's own name.

- **SaaS cookie capture.** Another folder targets a collaborative design SaaS platform. Headless Chrome 149 scripts log in, call the platform's API and save cookie jars, including `playwright_cookies.json`. An API call log (84,445 bytes) records requests sent with the client hint `sec-ch-ua: "HeadlessChrome";v="149"`.

The folders also hold seven `nuclei` output logs and two `subfinder` staging-subdomain lists of 227,107 bytes each. Both target brands, their domains and the custom script names are withheld.

CHAPTER 04

Stealers and Clippers

IN SHORT

Aurora, StealC, RedLineStealer and LaplasClipper supply the browser data and wallet theft. Of their 45 endpoints, 11 are confirmed C2 servers, and two suspected mail servers share one TLS fingerprint.

INSIDE THIS CHAPTER

How the stealers work 14

Confirmed stealer C2 servers 15

A shared mail fingerprint across two families 15

The LaplasClipper cluster 16

04 Stealers and Clippers

How the stealers work

Commodity stealers collect Chromium `Login Data`, `Cookies`, `Web Data` and wallet extension folders (MetaMask, Phantom and TronLink, for example) in memory. They package the result and send it over HTTP or TCP. A clipper module can stay behind to watch the clipboard and swap any copied wallet address for the attacker's own.

Figure 3 lays these steps out as a model sequence, from loader delivery to cookie theft to headless replay.

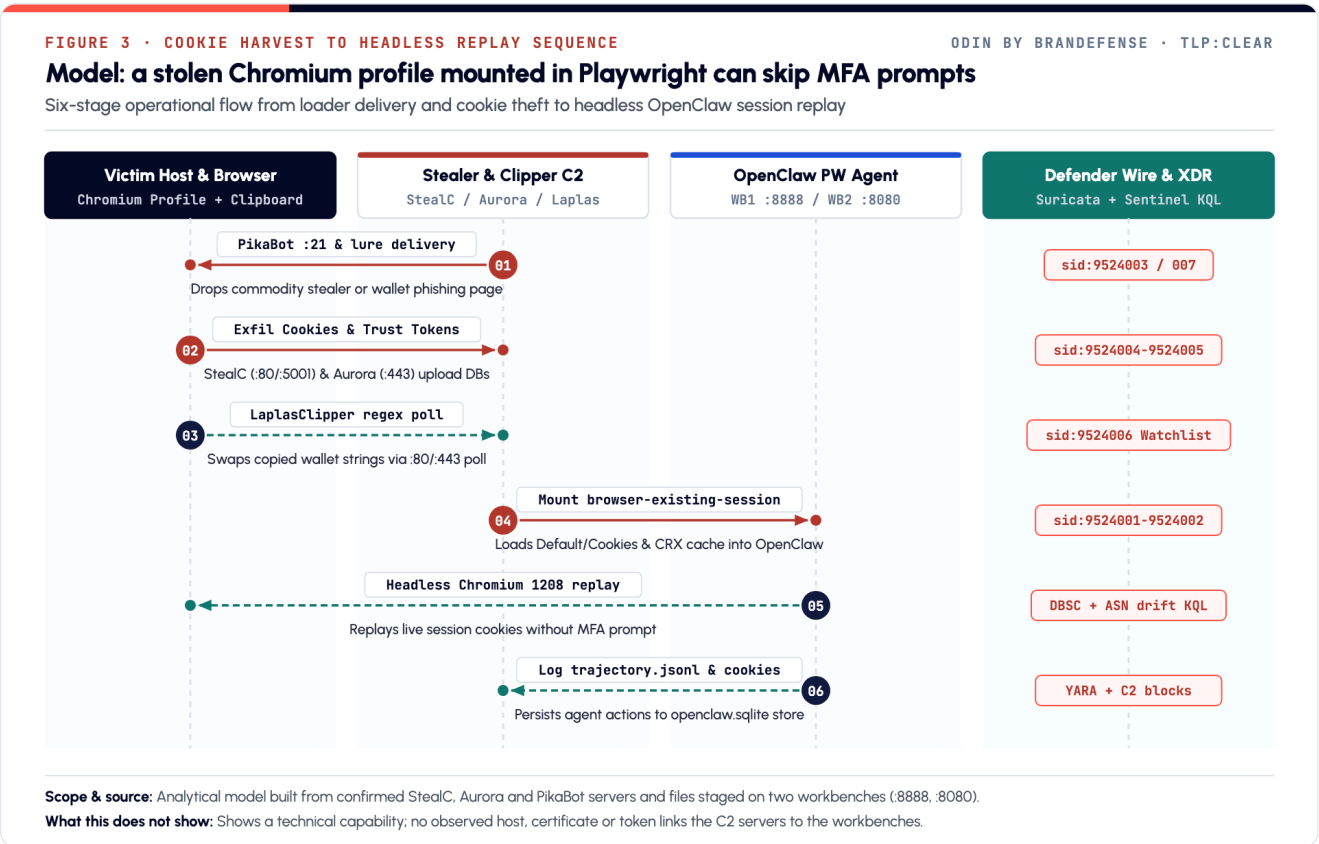


Figure 3: Model sequence from loader delivery and cookie theft to headless Playwright replay

Family	Endpoints	Hosts	Confirmed	Notes
AuroraStealer	13	7	6	Go stealer; one German host runs three C2 ports
Aurora	5	2	0	Suspected staging on ports 443, 3000 and 8080, seen 17 and 18 November
StealC	7	4	2	HTTP panels; one suspected 4-port mail host
RedLineStealer	6	6	3	Two Dutch hosts share one Windows certificate
LaplasClipper	14	14	0	All suspected, all on one US provider

Confirmed stealer C2 servers

AuroraStealer is written in Go, which makes it easy to run several listeners on one server. The German host 49.12.203[.]54 on Hetzner exposed three confirmed C2 ports on 13 November 2026: 21, 443 and 8443. Its certificate name is withheld.

StealC has two confirmed HTTP panels, on DigitalOcean in Canada and Google Cloud in Singapore. RedLineStealer has one listener on port 5800 on Hetzner in Finland. Two more RedLine hosts on RootLayer Web Services in the Netherlands present the same Windows certificate, `CN=WIN-UV1QM7FUHOK`, on port 3389.

Endpoint	Family	Country and network	Certificate or service
135.181.107[.]76:443	AuroraStealer	FI · Hetzner Online GmbH	`CN=ubuntu-4gb-hel1-7`
49.12.203[.]54:443	AuroraStealer	DE · Hetzner Online GmbH	TLS C2, certificate name withheld
49.12.203[.]54:8443	AuroraStealer	DE · Hetzner Online GmbH	Alternate TLS C2
49.12.203[.]54:21	AuroraStealer	DE · Hetzner Online GmbH	FTP listener
191.101.130[.]41:3389	AuroraStealer	US · Majestic Hosting Solutions	`CN=WIN-MG87181M2AR`
45.144.31[.]252:3389	AuroraStealer	RU · UFO Hosting LLC	`CN=VM-ba764849-...`
104.248.107[.]188:80	StealC	CA · DigitalOcean, LLC	HTTP panel
34.87.31[.]79:5001	StealC	SG · Google LLC	HTTP C2 on 5001
135.181.173[.]163:5800	RedLineStealer	FI · Hetzner Online GmbH	TCP C2 on 5800
185.222.58[.]51:3389	RedLineStealer	NL · RootLayer Web Services	`CN=WIN-UV1QM7FUHOK`
45.137.22[.]249:3389	RedLineStealer	NL · RootLayer Web Services	`CN=WIN-UV1QM7FUHOK`

The two RedLine hosts also share the JARM value `2ad2ad16d2ad2ad22c2ad2ad2ad2adfd9c9d14e4f4f67f94f0359f8b28f532`. JARM is a TLS server fingerprint. Many Windows servers share this kind of value, so pair it with the certificate name before acting.

A shared mail fingerprint across two families

Two suspected servers, one per family, present the exact same JARM on their IMAPS (993) and POP3S (995) ports: `0003fd00021d21d00042d43d000000e1eb348128a1eb0eda1c719dd71f6145`.

Host	Family	Country	Open ports
Suspected server A	AuroraStealer	RU, VPS provider	443, 995, 110, 993, 143
Suspected server B	StealC	NL, VPS provider	995, 110, 993, 143

The most likely explanation is that both servers were built from the same VPS control-panel mail template. That is a hosting link, not proof of a shared operator. Both addresses are withheld.

The LaplasClipper cluster

All 14 LaplasClipper endpoints sit with one US provider: 10 on port 80 and 4 on port 443. The provider runs managed DNS and Anycast HTTP redirection ranges, which many unrelated customers share. So all 14 stay suspected and on watchlists. Detection relies on the clipper's polling pattern (SID 9524006) rather than on blocking these addresses.

05

CHAPTER 05

Loaders and RAT Backdoors

IN SHORT

Loaders and backdoors make up 148 of the 193 endpoints across six families. Nine are confirmed C2 servers; four of them, a ChaluboRAT set in Hong Kong, are published for blocking.

INSIDE THIS CHAPTER

ChaluboRAT: four nodes in one Hong Kong /24 19

PikaBot and Ramnit: confirmed but withheld 19

Uniform suspected clusters 19

05 Loaders and RAT Backdoors

Stealers and clippers need a way in. Phishing loaders (PikaBot, TitanOmega), a long-running dropper network (Ramnit) and multi-platform RATs (ChaluboRAT, RaptorRAT, BitRAT) provide delivery and persistence. Figure 4 shows the port and service layout of the two workbenches and the main multi-port C2 hosts.

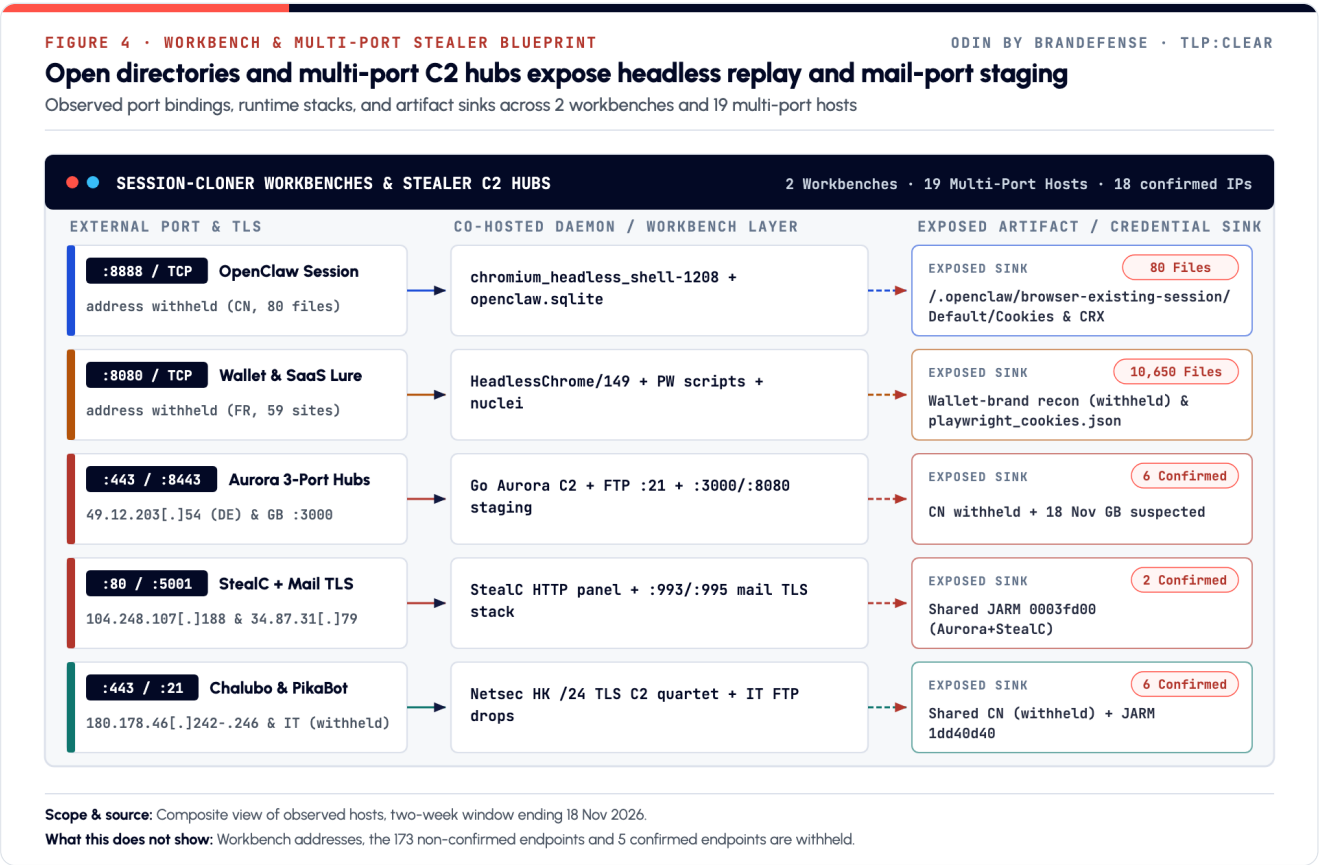


Figure 4: Open directories and multi-port C2 hubs expose headless replay and mail-port staging

Family	Endpoints	Hosts	Confirmed	Pattern
Ramnit	110	98	3	Mostly old, uncorroborated records; 5 on dynamic telecom networks
PikaBot	11	8	2	Two FTP drops in Italy; 9 suspected HTTP and HTTPS relays
RaptorRAT	9	9	0	All on Amazon AWS port 443, one shared JARM
TitanOmega	8	8	0	All on Liquid Web port 8080
ChaluboRAT	7	7	4	Four Hong Kong nodes in one `/24`; three port 8853 candidates
BitRAT	3	2	0	One HTTPS node and one mail-port host

ChaluboRAT: four nodes in one Hong Kong /24

ChaluboRAT targets Linux cloud servers. Four nodes in one Hong Kong `/24` on Netsec Limited are confirmed C2: 180.178.46[.]242, .244, .245 and .246, all on port 443. They share one certificate name (withheld) and the JARM `1dd40d40d00040d00042d43d000000e1ea2a807a629b496b664cf07ad7c08d`. Three more candidates on port 8853, in Romania, Moldova and Ukraine, stay on watchlists.

Endpoint	Family	Country and network	Seen
180.178.46[.]242:443	ChaluboRAT	HK · Netsec Limited	13 Nov 2026
180.178.46[.]244:443	ChaluboRAT	HK · Netsec Limited	13 Nov 2026
180.178.46[.]245:443	ChaluboRAT	HK · Netsec Limited	13 Nov 2026
180.178.46[.]246:443	ChaluboRAT	HK · Netsec Limited	13 Nov 2026

PikaBot and Ramnit: confirmed but withheld

PikaBot has two confirmed FTP drops on port 21 in Italy. Both sit on a shared web-hosting provider, where one address can serve many unrelated sites. Ramnit has three confirmed C2 endpoints: one TLS listener on port 8443 on a Thai broadband network, and two FTP listeners on port 21 on two Chinese networks.

We count these five as confirmed but do not publish their addresses. Shared hosting, broadband ranges and FTP listeners on infected machines can all belong to people other than the operator. Blocking them risks hitting third parties.

Uniform suspected clusters

PikaBot's nine suspected relays include three dual-port hosts (80 and 443) on Akamai and Amazon. TitanOmega binds port 8080 on all eight of its instances on Liquid Web in the United States. RaptorRAT runs on Amazon AWS only (7 in Germany, 2 in the United States), all on port 443 with the JARM `1dd40d40d00040d00042d43d000000831b6af40378e2dd35eeac4e9311926e`.

BitRAT pairs one HTTPS node in the Netherlands with a dual-port mail host (465 and 587) in Romania. Uniform clusters like these are good hunting leads. On shared cloud networks they are not safe to block without a second signal.

CHAPTER 06

Separating Signal from Noise

IN SHORT

Commodity malware feeds are noisy. Ramnit alone accounts for 110 endpoints, and Playwright and subdomain tools are also used by honest QA and red teams. Only 20 of 193 endpoints (10.4%) are confirmed.

INSIDE THIS CHAPTER

Four questions before you block 21

MITRE ATT&CK mapping 22

06 Separating Signal from Noise

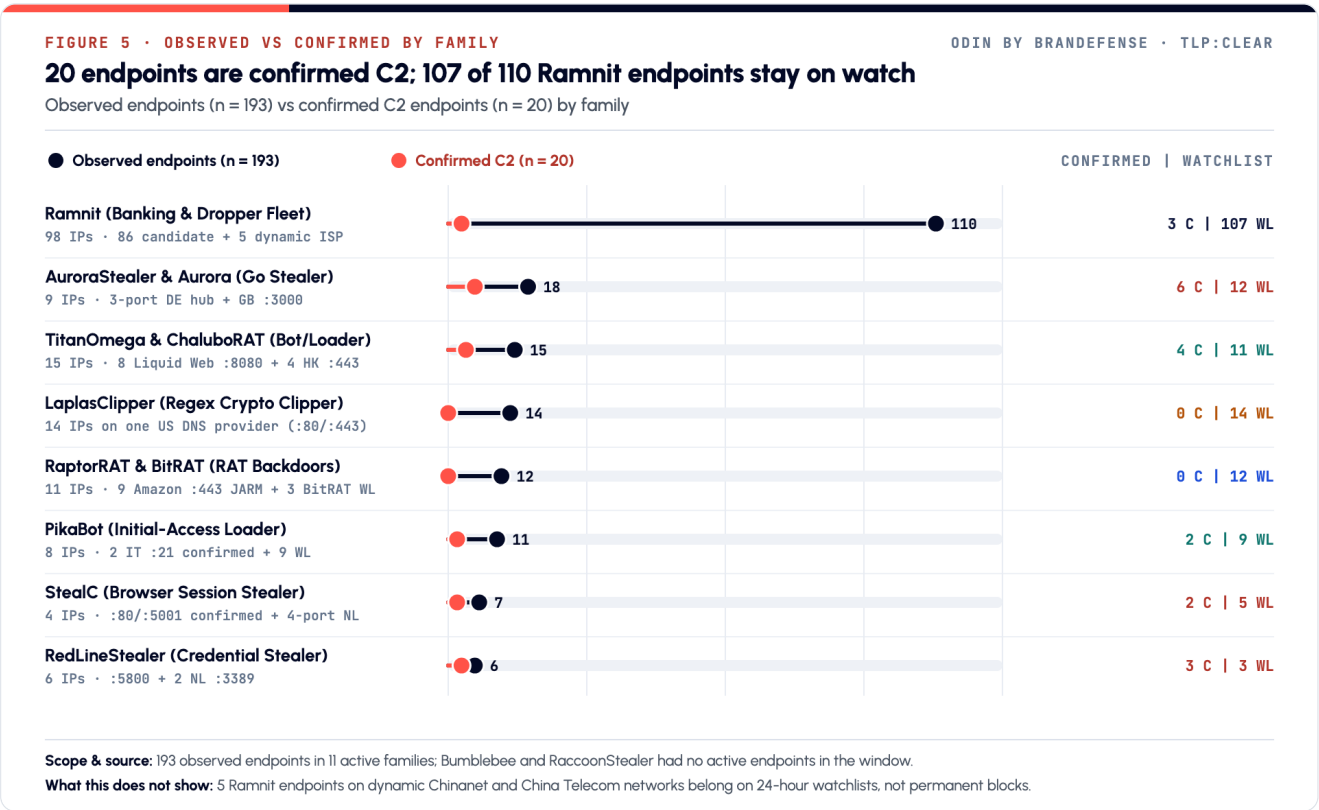


Figure 5: 20 confirmed C2 endpoints among 193 observed; most Ramnit endpoints stay on watchlists

Status	Endpoints	Network	Action
Confirmed C2	20 (on 18 IPs)	Mixed hosting	Block the 15 published endpoints; review blocks after 30 days
Suspected	77	Shared cloud, CDN or DNS networks	Watchlist; look for a second signal
Dynamic telecom	5	Ramnit endpoints on Chinanet and China Telecom subscriber ranges	24-hour watchlist
Candidate	91	Mixed, mostly older feed records	Hold until corroborated

Four questions before you block

Is it QA automation or session cloning?

Engineering teams run headless Chromium for end-to-end tests every day. A clean CI runner starts a fresh, empty browser context. It does not load a persistent `browser-existing-session/Default/Cookies` store next to chat-bot plugins. Allowlist your registered QA runners by exact hostname.

Is it a live server or an old record?

The 91 candidates include 86 Ramnit records, three ChaluboRAT nodes on port 8853 and two RedLineStealer nodes on ports 55615 and 1200. Some show corporate firewall certificates such as `CN=SANGFOR` and `CN=FortiGate` on reused addresses. These are not block-ready.

Is the address dynamic?

One Ramnit FTP node with a protocol match and four suspected Ramnit endpoints sit on dynamic Chinanet and China Telecom subscriber networks. Addresses there change hands often. A permanent block can cut off an unrelated customer later.

Is the network shared?

The 14 LaplasClipper nodes, 9 RaptorRAT nodes on Amazon and 8 TitanOmega nodes on Liquid Web sit on shared networks. They stay suspected until a payload or protocol match confirms them.

MITRE ATT&CK mapping

Technique	Where it shows up
T1539 Steal Web Session Cookie	Stealers copy Chromium `Cookies` and `Trust Tokens`
T1550.004 Use Alternate Authentication Material: Web Session Cookie	Headless Playwright loads a logged-in profile
T1115 Clipboard Data	LaplasClipper swaps copied wallet addresses

07

CHAPTER 07

What Defenders Should Do

IN SHORT

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

07 What Defenders Should Do

Within 24 hours

- [ ] Block the 15 published confirmed C2 endpoints (13 IPs) from Appendix A at the perimeter, and review them after 30 days.
- [ ] Put the 5 dynamic-telecom Ramnit endpoints and other dynamic addresses on 24-hour watchlists, not permanent blocks.
- [ ] Load Suricata SIDs 9524001 to 9524008 in alert mode and fill the KQL allowlist with exact QA hostnames.

Within 72 hours

- [ ] Turn on Chrome App-Bound Encryption on Windows endpoints.
- [ ] Alert when any process other than an installed browser reads Chromium `Default\Network\Cookies`, `Trust Tokens` or `Local Extension Settings`.
- [ ] Revoke sessions when a token suddenly appears from a new ASN or a headless client.

Within 7 to 30 days

- [ ] Pilot Device Bound Session Credentials (DBSC) or continuous access evaluation for high-value SaaS and cloud consoles.
- [ ] Audit external DNS and remove dangling staging `CNAME` records that a wallet or login lure could use.
- [ ] Restrict browser extension sideloading by enterprise policy.
- [ ] Require a second, hardware-backed confirmation for corporate crypto treasury transfers, so a swapped address is caught before it is sent.

08

CHAPTER 08

Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

How the Suricata signatures work 26

How the other rules work 26

08 Detection Rules

The full rules are in Appendix B. Each file targets a signal from this report. Every rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma, Kusto.Language); the rules were not replayed against live traffic.

Rule file	What it catches
`odin_playwright_session_cloners_and_stealers.rules`	Suricata 7: eight signatures for exposed OpenClaw session folders, Playwright cookie jars, stealer uploads, clipper polling and loader or RAT check-ins
`sentinel_openclaw_session_clone_and_stealer_hunt.kql`	Microsoft Sentinel and Defender XDR: contact with the published C2 addresses, joined with headless-browser and cookie-store process activity
`sigma_playwright_session_cloner_and_clipper_activity.yml`	Sigma (3 rules): profile mounts by headless runtimes, non-browser access to Chromium vaults, and clipboard-swap command lines
`odin_openclaw_aurora_laplas_stealc_artifacts.yar`	YARA (4 rules): OpenClaw session configs, Playwright cookie-capture scripts, Aurora and StealC vault manifests, and LaplasClipper regex configs

How the Suricata signatures work

SID	Looks for	Why
9524001	An HTTP 200 response on ports 8888, 8080 or 3000 whose body lists `.openclaw/browser-existing-session/` and `component_crx_cache`	Flags an exposed session-cloning directory
9524002	Outbound `POST /api/openclaw/session` with `browser-existing-session` and `trajectory` in the body	Catches an agent syncing replay logs
9524003	A request for `/playwright_cookies.json` with a `HeadlessChrome/` user agent	Catches cookie-jar transfer from a headless browser
9524004	Outbound `POST /aurora/gate` with `BrowserCookies` and `CryptoWallets` in the body	Catches AuroraStealer exfiltration (T1539)
9524005	Outbound multipart `POST` to a `.php` path with `cookies_list` and `Default\Network\Cookies`	Catches StealC cookie and LevelDB uploads
9524006	A request to `/bot/regex` with an `X-Laplas-Key:` header	Catches LaplasClipper polling for swap rules (T1115)
9524007	Outbound `POST /api/loader/stage` with `build_guid` in the body	Catches PikaBot and TitanOmega stage check-ins
9524008	Outbound `POST /gate/chalubo` with a user agent starting `Chalubo-Node/`	Catches ChaluboRAT heartbeats

Exempt named QA Playwright subnets from SID 9524003 if your test suites use a fixture called `playwright_cookies.json`. Send SID 9524001 hits to analyst review when they come from threat-intelligence sandboxes that browse open directories.

How the other rules work

The KQL query looks back 7 days. It carries the 13 published confirmed C2 addresses. Any device that contacts one of them always surfaces, whatever its name. The host allowlist applies only to the heuristic branch: `node`, `python`, `chrome-headless-shell`, `powershell.exe` or `rundll32.exe` talking on stealer and workbench ports.

That network view is joined with process events that mention `.openclaw`, `browser-existing-session`, `chromium_headless_shell`, `playwright_cookies.json`, `Default\Network\Cookies`, `Local Extension Settings` or `laplas`. A device also surfaces when it reaches 12 or more distinct remote IPs on those ports. Raise that threshold on Node build servers.

The first Sigma rule needs a headless runtime (`chrome-headless-shell`, `chromium`, `node` or `python3`) and a command line naming `.openclaw/browser-existing-session` or `playwright_cookies.json`. The second flags Windows command lines that touch Chromium cookie, token or extension paths, unless the process is an installed browser at its exact Program Files path. The third flags LaplasClipper polling strings or `Get-Clipboard` near a Bitcoin or Ethereum address pattern.

All Sigma allowlists use exact hostnames or exact executable paths. Per-user browser installs outside Program Files will show up and need local tuning.

YARA rule	Fires when
`ODIN_OpenClaw_Playwright_Session_Cloner_Config`	An OpenClaw session path or `openclaw.sqlite` appears with a Playwright headless-shell path, `component_crx_cache` or a `.trajectory.jsonl` name
`ODIN_Playwright_Cookie_Capture_Script`	`playwright_cookies.json` appears with `HeadlessChrome/` or the Playwright cookie APIs `context.cookies(` or `storageState`
`ODIN_Aurora_StealC_Browser_Vault_Manifest`	`/aurora/gate`, `BrowserCookies` and `CryptoWallets` appear together, or `cookies_list` appears with `Default\Network\Cookies`
`ODIN_LaplasClipper_Regex_Wallet_Swap_Config`	A LaplasClipper header or poll path appears with a BTC, ETH or TRX regex rule name

All four YARA rules stop at files under 512 KB.

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Published confirmed C2 endpoints 29

09 Indicators

Appendix A holds 15 published indicators: 15 confirmed C2 endpoints (IP and port) on 13 distinct addresses, and no file hashes. Five more confirmed endpoints, all suspected and candidate endpoints, both workbench addresses and every workbench file hash are withheld.

Type	Count
Confirmed C2 endpoints published for blocking (IP and port)	15 (13 IPs)
Confirmed C2 endpoints withheld (shared hosting, broadband or possibly infected hosts)	5
Representative watchlist entries withheld (2 workbenches, 5 suspected cluster nodes, 1 dynamic-ISP node); the other non-confirmed endpoints are withheld too	8
File hashes	0 published

Supporting fingerprints, text only:

- JARM `2ad2ad16d2ad2ad22c2ad2ad2ad2adfd9c9d14e4f4f67f94f0359f8b28f532` (RedLineStealer on 3389, pair with `CN=WIN-UV1QM7FUHOK`)
- JARM `1dd40d40d00040d00042d43d000000e1ea2a807a629b496b664cf07ad7c08d` (ChaluboRAT on 443)
- JARM `0003fd00021d21d00042d43d000000e1eb348128a1eb0eda1c719dd71f6145` (suspected Aurora and StealC mail ports)
- JARM `1dd40d40d00040d00042d43d000000831b6af40378e2dd35eeac4e9311926e` (suspected RaptorRAT on Amazon 443)
- Certificate names `CN=ubuntu-4gb-hel1-7`, `CN=WIN-MG87181M2AR` and `CN=WIN-UV1QM7FUHOK` on confirmed hosts

Published confirmed C2 endpoints

Endpoint	Family	Country	Network	Seen
135.181.107[.]76:443	AuroraStealer	FI	Hetzner Online GmbH	13 Nov 2026
191.101.130[.]41:3389	AuroraStealer	US	Majestic Hosting Solutions	13 Nov 2026
45.144.31[.]252:3389	AuroraStealer	RU	UFO Hosting LLC	13 Nov 2026
49.12.203[.]54:21	AuroraStealer	DE	Hetzner Online GmbH	13 Nov 2026
49.12.203[.]54:443	AuroraStealer	DE	Hetzner Online GmbH	13 Nov 2026
49.12.203[.]54:8443	AuroraStealer	DE	Hetzner Online GmbH	13 Nov 2026
180.178.46[.]242:443	ChaluboRAT	HK	Netsec Limited	13 Nov 2026
180.178.46[.]244:443	ChaluboRAT	HK	Netsec Limited	13 Nov 2026
180.178.46[.]245:443	ChaluboRAT	HK	Netsec Limited	13 Nov 2026
180.178.46[.]246:443	ChaluboRAT	HK	Netsec Limited	13 Nov 2026
135.181.173[.]163:5800	RedLineStealer	FI	Hetzner Online GmbH	13 Nov 2026

Endpoint	Family	Country	Network	Seen
185.222.58[.]51:3389	RedLineStealer	NL	RootLayer Web Services	13 Nov 2026
45.137.22[.]249:3389	RedLineStealer	NL	RootLayer Web Services	13 Nov 2026
104.248.107[.]188:80	StealC	CA	DigitalOcean, LLC	13 Nov 2026
34.87.31[.]79:5001	StealC	SG	Google LLC	13 Nov 2026

VPS leases change hands after takedowns. Re-check each address before renewing a block after 30 days.

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

- **What we saw.** We saw servers, open ports, certificates and files in open directories. We did not see any victim machine, any replayed session or any loss on the target side.
- **The model is a model.** The four-part chain in Chapter 2 explains how the pieces could fit. Nothing we observed links a given stealer server to either workbench.
- **Workbench owners are unknown.** Workbench 1 could hold stolen browser data or its owner's own profile. Workbench 2 could belong to an attacker or to an unhardened security contractor.
- **Confirmed is not complete.** Twenty confirmed endpoints are a floor. Many of the 77 suspected and 91 candidate endpoints may be real C2 servers that lacked a second signal.
- **Family names.** Family names describe the malware, not a named threat group or operator.
- **Dates.** Seen dates are when we observed an endpoint. Older dates on some Ramnit records are record dates, not the length of this study.
- **Third parties.** Targeted wallet and SaaS brands, both workbench addresses and five confirmed endpoints on shared or subscriber networks are withheld.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. Google Chromium Security, Improving the Security of Chrome Cookies on Windows: security.googleblog.com/2024/07/improving-security-of-chrome-cookies-on.html (verified on 18 November 2026).
2. Microsoft Playwright Documentation, BrowserType.launchPersistentContext: playwright.dev/docs/api/class-browsertype (verified on 18 November 2026).
3. W3C Web Application Security Working Group, Device Bound Session Credentials: w3c.github.io/webappsec-dbsc (verified on 18 November 2026).
4. MITRE ATT&CK, Steal Web Session Cookie (T1539): attack.mitre.org/techniques/T1539 (verified on 18 November 2026).
5. CISA, Implementing Phishing-Resistant MFA: www.cisa.gov/resources-tools/resources/implementing-phishing-resistant-mfa (verified on 18 November 2026).
6. OISF, Suricata 7.0 User Guide: HTTP Keywords: docs.suricata.io/en/suricata-7.0.0/rules/http-keywords.html (verified on 18 November 2026).
7. Salesforce Engineering, Easily Identify Malicious Servers on the Internet with JARM: engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a (verified on 18 November 2026).

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

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

INSIDE

A. Indicators of compromise (15) 36

B. Detection rules (4 files) 37

A Indicators of Compromise

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

ID	Type	Indicator	Port	Context	Action
R24-IOC-01	IPv4	135.181.107[.]76	443	aurorastealer	BLOCK
R24-IOC-02	IPv4	191.101.130[.]41	3389	aurorastealer	BLOCK
R24-IOC-03	IPv4	45.144.31[.]252	3389	aurorastealer	BLOCK
R24-IOC-04	IPv4	49.12.203[.]54	21	aurorastealer	BLOCK
R24-IOC-05	IPv4	49.12.203[.]54	443	aurorastealer	BLOCK
R24-IOC-06	IPv4	49.12.203[.]54	8443	aurorastealer	BLOCK
R24-IOC-07	IPv4	180.178.46[.]242	443	chaluborat	BLOCK
R24-IOC-08	IPv4	180.178.46[.]244	443	chaluborat	BLOCK
R24-IOC-09	IPv4	180.178.46[.]245	443	chaluborat	BLOCK
R24-IOC-10	IPv4	180.178.46[.]246	443	chaluborat	BLOCK
R24-IOC-11	IPv4	135.181.173[.]163	5800	redlinestealer	BLOCK
R24-IOC-12	IPv4	185.222.58[.]51	3389	redlinestealer	BLOCK
R24-IOC-13	IPv4	45.137.22[.]249	3389	redlinestealer	BLOCK
R24-IOC-14	IPv4	104.248.107[.]188	80	stealc	BLOCK
R24-IOC-15	IPv4	34.87.31[.]79	5001	stealc	BLOCK

B Detection Rules

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

odin_openclaw_aurora_laplas_stealc_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Monograph 24 / 30 YARA Artifact Suite
   Scope: OpenClaw Playwright session-cloning configs, Playwright cookie-capture scripts,
          AuroraStealer/StealC exfiltration manifests, and LaplasClipper regex configs.
   Classification: TLP:CLEAR
*/

rule ODIN_OpenClaw_Playwright_Session_Cloner_Config
{
    meta:
        author = "Odin by Brandefense"
        monograph = "24 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt"
        description = "Matches OpenClaw headless Playwright existing-browser-session state and trajectory
configs"
    strings:
        $oc1 = ".openclaw/browser-existing-session" ascii
        $oc2 = "openclaw.sqlite" ascii
        $pw1 = "chromium_headless_shell-" ascii
        $pw2 = "component_crx_cache" ascii
        $tr1 = ".trajectory.jsonl" ascii
    condition:
        filesize < 512KB and ($oc1 or $oc2) and ($pw1 or $pw2 or $tr1)
}

rule ODIN_Playwright_Cookie_Capture_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "24 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt"
        description = "Matches Playwright scripts or logs that write an authenticated cookie jar from a headless
Chrome session"
    strings:
        $ck1 = "playwright_cookies.json" ascii
        $ua1 = "HeadlessChrome/" ascii
        $pa1 = "context.cookies(" ascii
        $pa2 = "storageState" ascii
    condition:
        filesize < 512KB and $ck1 and ($ua1 or $pa1 or $pa2)
}

rule ODIN_Aurora_StealC_Browser_Vault_Manifest
{
    meta:
        author = "Odin by Brandefense"
        monograph = "24 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt"
        description = "Matches AuroraStealer Go JSON gate configs and StealC browser cookie/vault multipart
manifests"
    strings:
        $au1 = "/aurora/gate" ascii
        $au2 = "BrowserCookies" ascii
        $au3 = "CryptoWallets" ascii
        $st1 = "cookies_list" ascii
        $st2 = "Default\\Network\\Cookies" ascii
    condition:
```

```
        filesize < 512KB and (($au1 and $au2 and $au3) or ($st1 and $st2))
}

rule ODIN_LaplasClipper_Regex_Wallet_Swap_Config
{
    meta:
        author = "Odin by Brandefense"
        monograph = "24 / 30"
        classification = "TLP:CLEAR"
        usage = "hunt"
        description = "Matches LaplasClipper HTTP regex polling headers and multi-chain wallet substitution
manifests"
    strings:
        $lp1 = "X-Laplas-Key" ascii
        $lp2 = "/bot/regex" ascii
        $rx1 = "btc_bech32_regex" ascii nocase
        $rx2 = "eth_erc20_regex" ascii nocase
        $rx3 = "trx_trc20_regex" ascii nocase
    condition:
        filesize < 512KB and ($lp1 or $lp2) and ($rx1 or $rx2 or $rx3)
}
```

odin_playwright_session_cloners_and_stealers.rules

SURICATA

```
# Odin by Brandefense - Monograph 24 / 30 Suricata 7.x Detection Suite
# Coverage: OpenClaw Playwright session cloning, Playwright cookie capture, AuroraStealer, StealC, LaplasClipper,
PikaBot, ChaluboRAT
# Classification: TLP:CLEAR
# Loads in Suricata 7.0.12; not yet replayed against captured traffic from these families. Tune before inline
blocking.

alert http $EXTERNAL_NET [8888,8080,3000] -> $HOME_NET any (msg:"ODIN [R24] OpenClaw Existing-Browser-Session
Directory Listing in HTTP Response"; flow:established,to_client; http.stat_code; content:"200";
http.response_body; bsize:>36; content:".openclaw/browser-existing-session/"; fast_pattern;
content:"component_crx_cache"; distance:0; classtype:policy-violation; sid:9524001; rev:2; metadata:created_at
2026_11_18, updated_at 2026_11_20, deployment perimeter, policy open_directory_exposure;)

alert http $HOME_NET any -> $EXTERNAL_NET [8888,8080,3000] (msg:"ODIN [R24] OpenClaw Headless Chromium Session-
Replay Trajectory Sync"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/
openclaw/session"; fast_pattern; http.request_body; bsize:>32; content:"\"browser-existing-session\"";
content:"\"trajectory\""; distance:0; classtype:trojan-activity; sid:9524002; rev:2; metadata:created_at
2026_11_18, updated_at 2026_11_20, deployment perimeter, tactic credential_access;)

alert http $HOME_NET any -> $EXTERNAL_NET [80,443,8080] (msg:"ODIN [R24] Playwright Cookie-Jar Upload or Fetch
from Headless Chrome"; flow:established,to_server; http.uri; content:"/playwright_cookies.json"; fast_pattern;
http.user_agent; content:"HeadlessChrome/"; classtype:trojan-activity; sid:9524003; rev:2; metadata:created_at
2026_11_18, updated_at 2026_11_20, deployment perimeter, tactic credential_access;)

alert http $HOME_NET any -> $EXTERNAL_NET [443,8443,8080,3000] (msg:"ODIN [R24] AuroraStealer Go Binary Browser
Cookie and Crypto-Vault Exfiltration"; flow:established,to_server; http.method; content:"POST"; http.uri;
content:"/aurora/gate"; fast_pattern; http.request_body; bsize:>34; content:"\"BrowserCookies\"";
content:"\"CryptoWallets\""; distance:0; classtype:trojan-activity; sid:9524004; rev:2; metadata:created_at
2026_11_18, updated_at 2026_11_20, deployment perimeter, tactic exfiltration;)

alert http $HOME_NET any -> $EXTERNAL_NET [80,5001,8080] (msg:"ODIN [R24] StealC Chromium Cookies and Extension
LevelDB Multipart Upload"; flow:established,to_server; http.method; content:"POST"; http.uri; content:".php";
http.request_body; bsize:>40; content:"name=\"cookies_list\""; fast_pattern; content:"Default\\Network\\Cookies";
distance:0; classtype:trojan-activity; sid:9524005; rev:2; metadata:created_at 2026_11_18, updated_at 2026_11_20,
deployment perimeter, tactic credential_access;)

alert http $HOME_NET any -> $EXTERNAL_NET [80,443] (msg:"ODIN [R24] LaplasClipper Regex Cryptocurrency Clipboard
Address-Swap Poll"; flow:established,to_server; http.uri; content:"/bot/regex"; fast_pattern; http.header;
bsize:>16; content:"X-Laplas-Key|3A|"; classtype:trojan-activity; sid:9524006; rev:2; metadata:created_at
2026_11_18, updated_at 2026_11_20, deployment perimeter, tactic collection;)

alert http $HOME_NET any -> $EXTERNAL_NET [80,443,8080] (msg:"ODIN [R24] PikaBot and TitanOmega Initial-Access
Loader Stage Check-In"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/loader/
stage"; fast_pattern; http.request_body; bsize:>24; content:"\"build_guid\""; classtype:trojan-activity; sid:
```

```
9524007; rev:2; metadata:created_at 2026_11_18, updated_at 2026_11_20, deployment perimeter, tactic
command_and_control;)

alert http $HOME_NET any -> $EXTERNAL_NET [80,82,443,8443,8853] (msg:"ODIN [R24] ChaluboRAT and Ramnit Persistent
Backdoor Node Heartbeat"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/gate/
chalubo"; fast_pattern; http.user_agent; bsize:>14; content:"Chalubo-Node/"; startswith; classtype:trojan-
activity; sid:9524008; rev:2; metadata:created_at 2026_11_18, updated_at 2026_11_20, deployment perimeter, tactic
command_and_control;)
```

sentinel_openclaw_session_clone_and_stealer_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 24 / 30 Microsoft Sentinel & Defender XDR Hunt
// Correlates egress to confirmed stealer/loader C2 servers with OpenClaw Playwright session cloning and Chromium
cookie access
// Classification: TLP:CLEAR
// The host allowlist applies only to the process and port heuristics; a match on a confirmed C2 address always
surfaces.

let Lookback = 7d;
// Placeholder host names: replace with your own approved hosts.
let ApprovedAutomationHosts = dynamic(["QA-PLAYWRIGHT-CI-01", "SYNTHETIC-MON-01", "SEC-LAB-RUNNER-01"]);
let ConfirmedStealerC2IPs = dynamic([
    "135.181.107.76", "191.101.130.41", "45.144.31.252", "49.12.203.54",
    "180.178.46.242", "180.178.46.244", "180.178.46.245", "180.178.46.246",
    "135.181.173.163", "185.222.58.51", "45.137.22.249",
    "104.248.107.188", "34.87.31.79"
]);
let SessionAndStealerPorts = dynamic([21, 80, 82, 443, 3000, 3389, 5001, 5800, 8080, 8443, 8853, 8888]);
let SuspiciousSessionProcs = DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAutomationHosts))
    | where ProcessCommandLine has_any (
        ".openclaw", "browser-existing-session", "chromium_headless_shell",
        "playwright_cookies.json",
        "Default\\Network\\Cookies", "Local Extension Settings", "laplas"
    )
    | project ProcTime = Timestamp, DeviceId, DeviceName, InitiatingProcessFileName, FileName,
ProcessCommandLine;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback)
| where RemoteIP in (ConfirmedStealerC2IPs)
    or (not(DeviceName in~ (ApprovedAutomationHosts))
        and RemotePort in (SessionAndStealerPorts)
        and InitiatingProcessFileName in~ ("node", "node.exe", "chrome-headless-shell", "python", "python3",
"powershell.exe", "rundll32.exe"))
| summarize
    NetFirstSeen = min(Timestamp),
    NetLastSeen = max(Timestamp),
    DistinctRemoteIPs = dcount(RemoteIP),
    DistinctPorts = make_set(RemotePort, 16),
    MatchedC2IPs = make_set_if(RemoteIP, RemoteIP in (ConfirmedStealerC2IPs), 20),
    ConnectionCount = count()
    by DeviceId, DeviceName, InitiatingProcessFileName
| join kind=leftouter (SuspiciousSessionProcs) on DeviceId, DeviceName
| where array_length(MatchedC2IPs) > 0 or isnotempty(ProcessCommandLine) or DistinctRemoteIPs >= 12
| extend AlertDisposition = case(
    array_length(MatchedC2IPs) > 0, "BLOCK_CONFIRMED_STEALER_OR_LOADER_C2",
    ProcessCommandLine has_any (".openclaw", "browser-existing-session", "playwright_cookies.json"),
"ISOLATE_HEADLESS_SESSION_CLONER_REPLAY",
    "HUNT_STEALER_OR_CLIPPER_BEACON"
)
| project NetFirstSeen, NetLastSeen, DeviceName, AlertDisposition, MatchedC2IPs, DistinctRemoteIPs,
DistinctPorts, InitiatingProcessFileName, ProcessCommandLine, ConnectionCount
| sort by ConnectionCount desc
```

sigma_playwright_session_cloner_and_clipper_activity.yml

SIGMA

```
title: Unapproved OpenClaw Headless Playwright Existing-Browser-Session Mount
id: 8c41e290-5d73-4b18-9f24-202611182401
status: experimental
description: Detects headless Chromium, Node or Python processes that mount an existing browser profile directory
(.openclaw/browser-existing-session) or read a Playwright cookie jar outside approved QA hosts.
author: Odin by Brandefense
date: 2026/11/18
modified: 2026/11/20
tags:
  - attack.defense_evasion
  - attack.lateral_movement
  - attack.t1550.004
logsource:
  category: process_creation
  product: linux
detection:
  selection_runtime:
    Image|endswith:
      - '/chrome-headless-shell'
      - '/chromium'
      - '/node'
      - '/python3'
  selection_session_args:
    CommandLine|contains:
      - '.openclaw/browser-existing-session'
      - 'playwright_cookies.json'
  filter_approved_qa:
    # Placeholder host names: replace with your own approved hosts.
    ComputerName:
      - 'QA-PLAYWRIGHT-CI-01'
      - 'SYNTHETIC-MON-01'
  condition: selection_runtime and selection_session_args and not filter_approved_qa
falsepositives:
  - Sanctioned end-to-end QA automation runners that are not yet in the exact-hostname allowlist
level: high
---
title: Non-Browser Process Harvesting Chromium Cookies and Extension LevelDB Vaults
id: 3f72a814-6c19-44e2-a108-202611182402
status: experimental
description: Identifies commodity infostealers (StealC, AuroraStealer, RedLineStealer) archiving or reading
Chromium Cookies, Trust Tokens, and Local Extension Settings vaults from processes other than installed browsers.
author: Odin by Brandefense
date: 2026/11/18
modified: 2026/11/20
tags:
  - attack.credential_access
  - attack.t1539
logsource:
  category: process_creation
  product: windows
detection:
  selection_vault_paths:
    CommandLine|contains:
      - '\User Data\Default\Network\Cookies'
      - '\User Data\Default\Local Extension Settings'
      - '\User Data\Default\Trust Tokens'
      - 'component_crx_cache'
  filter_legit_browsers:
    Image:
      - 'C:\Program Files\Google\Chrome\Application\chrome.exe'
      - 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe'
      - 'C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe'
      - 'C:\Program Files\Microsoft\Edge\Application\msedge.exe'
      - 'C:\Program Files\BraveSoftware\Brave-Browser\Application\brave.exe'
  filter_lab:
    # Placeholder host names: replace with your own approved hosts.
    ComputerName:
      - 'SEC-LAB-RUNNER-01'
```

```
  condition: selection_vault_paths and not filter_legit_browsers and not filter_lab
falsepositives:
  - Endpoint forensic collection scripts run by authorized incident responders
  - Per-user browser installs outside Program Files, which the exact-path allowlist does not cover
level: high
---
title: LaplasClipper Regex Cryptocurrency Clipboard Hijacker Execution
id: 9b15d407-2a84-47c1-8e63-202611182403
status: experimental
description: Detects execution patterns associated with LaplasClipper or PowerShell/Go clipboard monitors polling
regex rules to swap copied BTC, ETH, TRX, or SOL wallet addresses.
author: Odin by Brandefense
date: 2026/11/18
tags:
  - attack.collection
  - attack.t1115
logsource:
  category: process_creation
  product: windows
detection:
  selection_clipper:
    CommandLine|re: '(/bot/regex|X-Laplas-Key|Get-Clipboard.*(bc1[ac-hj-np-z02-9]|0x[a-fA-F0-9]{40}))'
  filter_lab:
    # Placeholder host names: replace with your own approved hosts.
    ComputerName:
      - 'SEC-LAB-RUNNER-01'
  condition: selection_clipper and not filter_lab
falsepositives:
  - Dedicated malware analysis sandboxes listed in asset inventory
level: high
```

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/session-cloners-and-stealer-fleets/
Short Read · brandefense.io/odin-research/blog/stolen-browser-sessions-headless-agents-and-the-stealers-that-feed/
Defense Kit · brandefense.io/odin-research/defense-kit/