

Section 5

The vulnerability profile of the pre-2023 API-key architecture

Investigated by: Agent Sofia

The vulnerability profile of the pre-2023 API-key architecture

One session. More access than the user sees.

Context & source limits

Valve's API-key architecture, as it stood before the 2023 changes, departed from baseline industry standards (OWASP ASVS, NIST SP 800-63B) and is the mechanism described in the automated inventory-theft complaints preserved in the evidence archive (Sections 6–8).

The rule Valve applied: the account holder created the key, accepted the terms, and bears the loss — that is the answer in the archived refusals. **The record:** until December 2023 a stolen session cookie alone could register that key. No password. No Steam Guard. No notification to the person whose account it was. **The user did not create the key, could not see it, and was told he did.** Mobile confirmation was added on 2023-12-04 — about seven years after the 2016 statement and roughly six after the scam panels the archive preserves. Those years were not paid by Valve. They were paid by the accounts emptied in the meantime, a share of them children's.

OWASP ASVS 2.8 / NIST SP 800-63B SEVERITY: CRITICAL

1. Insufficient Step-up Authentication

Steam allowed the generation of highly privileged, long-lived API keys using only a session cookie. No 2FA (Steam Guard) or step-up authentication challenge was required when setting up or modifying a developer API key, so the Steam Guard mobile confirmation — the only second factor Steam offers — did not protect this step. (A hardware second factor was requested in Valve's own tracker in 2016 and the issue is still open: Appendix D, GH-01.)

Status: unpatched before 2023. Allowed a script holding a stolen cookie to register a key without any user interaction.

OWASP A01:2021 / Least Privilege SEVERITY: CRITICAL

2. Lack of Principle of Least Privilege

The Steam Web API lacked any role-based authorisation scopes. A developer API key issued for "read-only" inventory status possessed complete "write" permissions — including the ability to manipulate existing trades, initiate new ones, register webhooks and decline valid pending exchanges.

Status: systemic design defect. One key = full read/write control of the account's inventory and trades.

OWASP A07:2021 / Sessions SEVERITY: HIGH

3. Broken Session Management

Hijacked web sessions originating from server-hosting IP ranges were reported by users to persist. Standard practice is to terminate sessions on geographic anomalies and on password change; user reports collected in the archive describe sessions that remained active after a password change.

Status: reported behaviour (user complaints in the archive). Sessions could generate API keys without re-authentication.

Domain-level blocking was technically feasible ENFORCEMENT FEASIBILITY

4. Domain-Level Blocking of Auth Hosts Was Technically Feasible

Valve already operates a domain list (the Link Filter) and already receives every auth hostname in its own OpenID requests. The two mechanisms are not connected in any way visible from outside: as of 2026-09-22, in 26 pairs the main domain was on the filter while the proxy carrying its login was not.

Connecting them is a small engineering task: read the `openid.realm` hostname from each incoming request, compare it against a list, and refuse or flag the request. The illustration on the right shows the shape of such a routine. Refusing the auth host — rather than only the public landing page — would interrupt the "Sign in with Steam" step that these platforms depend on.

No public evidence shows that Valve runs such a check. The Steam Community Market's combined fee of about 15% applies to sales completed inside the Market; it is not collected on these platforms' external trades.

```
block_auth_hosts.py ILLUSTRATION
// Read the host Steam already receives — no scanning
def on_openid_request(request):
    host = parse_openid_realm(request)
    if host in blocked_domains:
        refuse(request)
```

Note: pseudocode for illustration only. Both halves already exist at Valve — the auth hostname arrives in the request, and the domain list is the Link Filter. No public record explains why they are not connected.
Section 5.1 — The standards were not unpublished

Valve builds against these rules every day. It did not build to them here.

Steam ships on the App Store and Google Play, sells through the major card processors, and consumes third-party APIs like any other large platform. Each of those relationships carries a published, dated security requirement that Valve must satisfy to keep operating. Set the dates of those requirements against the dates on which Steam's own Web API acquired the same protections. **Nothing below is a private standard, an internal memo or a trade secret: every row is a public document with a publication date.**

Published external security requirements, with their dates, set against the date the Steam Web API acquired the equivalent control

The published standard, and the date it took effect	The Steam Web API on the same question
Scoped authorisationOAuth 2.0 — RFC 6749, October 2012. Section 3.3 defines scope so a client receives only the access it asks for. It has been the default of every major platform API since.	Granular scopes arrived on 16 January 2026. Before that a single Steam Web API key carried full read and write over the account's trades regardless of why it was issued. Thirteen years and three months after the specification that defines the control.
A key is not an authorisationGoogle's own Cloud documentation: "A standard API key doesn't authenticate a principal", and for APIs that create or manage resources, "don't use authorization keys in production."	The Steam Web API key was exactly that. A bearer string, issued to an account, sufficient on its own to read the inventory, cancel a pending trade and send a new one. No principal was authenticated at the point of use.
Multi-factor authentication on privileged accessPCI DSS 3.2, requirement 8.3 — mandatory from 1 February 2018 for all non-console administrative and all remote access to the cardholder data environment. Apple then required two-factor authentication of every Developer Program account holder from 27 February 2019: "developers with the Account Holder role in a developer program will need to enable two-factor authentication to sign in."	A second factor was required to create an API key from 4 December 2023. Until then a stolen session cookie was enough, with no prompt on the account holder's phone and no notification afterwards. Four years and nine months after Apple made Valve itself use a second factor merely to sign in to a developer portal.

The published standard, and the date it took effect

Credential lifecycle and revocation Twitch requires of every third-party application: “Your app must validate the OAuth token when it starts and on an hourly basis thereafter”, and states that it audits for applications that do not. Its legacy v5 API was retired outright on 28 February 2022 on a published schedule.

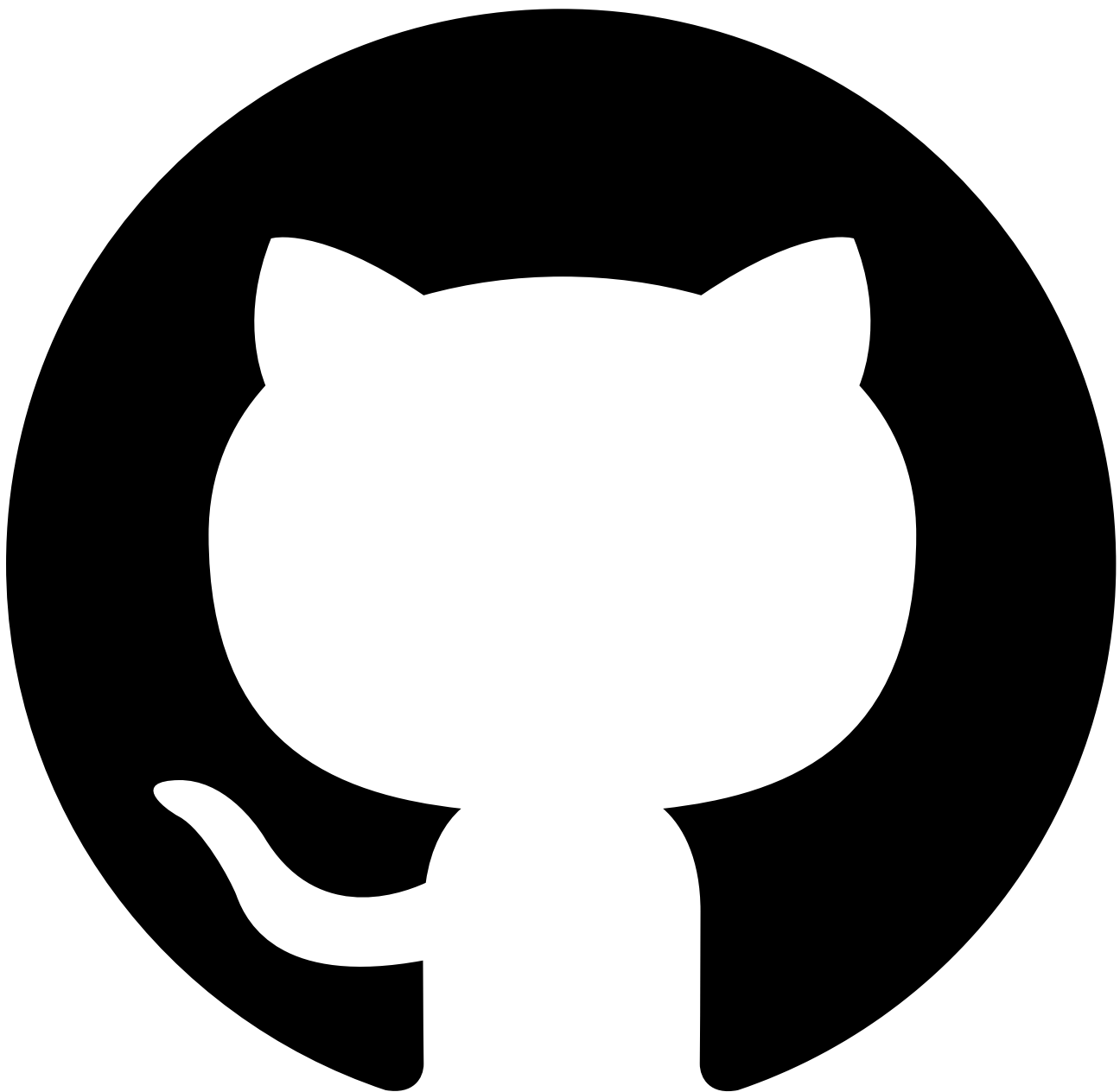
The Steam Web API on the same question

No published key-lifecycle record. The 2023 change protected the creation endpoint. No Valve statement located with it addresses keys already registered, and no changelog accompanied the change at all. Whether keys created before it were invalidated cannot be established from public data — which is the question a data subject would need answered to know whether a key created on their account years earlier is still live.

Architectural Benchmark // Token Architecture

GitHub Fine-Grained Scopes vs Valve Steam Web API

Valve developers manage open-source software on GitHub daily using granular least-privilege tokens. Compare that with the monolithic key issued to 130M Steam players.



GitHub Fine-Grained Token
Used by Valve on GitHub

Fine-grained token with scoped isolation and mandatory lifecycle enforcement.

Repository Access: Selected Repos only (1 repo)

Issue Permissions: Read-only [✓]

Code & Secrets: No access [✗]

Expiration: 30 Days (Mandatory TTL)

Audit Logging: IP, User-Agent & Timestamp

Security Posture: Least-privilege token limits damage to a single read-only resource.

! Valve Steam Web API Key

Issued to 130M Users

Single 32-character master string carrying all-or-nothing trade and inventory authority.

Account Access: FULL ACCOUNT (Monolithic)

Trade Management: Read, Cancel, Decline [!]

Granular Scopes: NONE (Arrived 2026, 13y late)

Expiration: NEVER (Valid indefinitely)

Audit Logging: NONE (Zero IP history for user)

Vulnerability: Stolen key allows silent interception of all outgoing trade offers.

The standard applied in one direction

Suppose a vendor Valve depends on — a cloud provider, a payment processor, a CDN — left a credential endpoint without a second factor, and months later Valve's own corporate data left through a key minted at that endpoint. Would Valve accept that the fault lay with whoever clicked the link? The archived refusals preserved in this dossier answer the mirror-image question for the user: the position taken is that the account holder created the key and bears the loss. **The same architecture is described as the vendor's failure in one direction and the user's responsibility in the other.** Which it is, is not a technical question, and it is not one this investigation can settle — it is what a court or a supervisory authority decides.

Illustration, not evidence — the standard read back the other way

- A payment processor that let a third-party script mint a full-privilege transaction credential from a browser cookie alone, with no prompt to the account holder.
- An identity provider that closed the hole its tokens were minted through and published nothing about the tokens already minted.
- A bank that answered a disputed transfer by observing that the credential used was registered on the customer's own account, and closed the ticket.

None of these is offered as a finding about any named company. They are the same design decisions, moved to a setting where the supervisory response is well documented — which is the comparison a regulator is being asked to make.

What the record establishes: the controls absent from the Steam Web API were public, dated and, in several cases, conditions Valve had to satisfy to keep shipping its own products. The interval between each standard and the equivalent Steam change is measurable and is set out above. **What it does not establish is why.** No Valve statement of reasons for any of these intervals has been located, and none is assumed here; that record exists inside Valve and has not been published.

Sources, each a public document: OAuth 2.0 — [RFC 6749, October 2012 ↗](#) · Google Cloud — [API keys documentation ↗](#) · PCI DSS 3.2 requirement 8.3 — [PCI SSC resource guide ↗](#) · Apple — [Upcoming Two-Factor Authentication Requirement for Account Holders, 20 February 2019 ↗ archived 2025-05-06 ↗](#) · Twitch — [token validation requirement ↗](#) and the [v5 shutdown notice ↗ archived 2025-01-27 ↗](#). Steam's

own dates are those in the update record in this section; the 2023 and 2026 entries are the ones Valve shipped without a changelog. Retrieved 2026-09-22.

Valve's own update record — dated, from public sources

Date	Valve change	Announced?	What it closed	Lag: mechanism public → fix	Source
2026-09-01	Automated Suspicious Chat Flagging	Yes	Dangerous links / patterns in group chats	—	news ↗
2026-01-16	Steamworks Web API granular scopes	Yes	Least-privilege scopes for developer keys	—	announcement ↗
2025-07-15	CS2 Trade Protection & reversal window	Yes	Nothing. Not a root-cause security fix. Support-side account recovery access remains untouched. Triggering reversal inflicts an automatic 30-day trade lockout (Skinvault ↗), while cash-trading shock caused a \$104M (-25%) 24h market cap plunge (CSMarketCap ↗). In 2026, timers became per-item (Skin.land ↗).	—	FAQ ↗
2025-05-14	Account Security Announcement	Yes	Stealer / session-hijack guidance	—	announcement ↗
2024-10-02	CS2 'The Armory' 7-day hold	Yes	Hold on trade/market for Armory items	—	announcement ↗
2024-09-18	New-device login trade restriction	Yes	2–7 day trade lock on new devices	—	FAQ ↗
2024-02-14	Profile name & avatar change hold	Yes	2–4 h cooldown on trades after rename	—	post ↗
2023-12-04	Mobile confirmation for API keys	No change	Silent cookie-only key registration	≈6 years after the 2017-era panels	dev page ↗
2023-10-24	SMS MFA for game build publications	Yes	Steamworks build publishing	—	announcement ↗
2023-07-12	Custom biography content warning	Yes	Phishing links in bios	—	forum ↗

Date	Valve change	Announced?	What it closed	Lag: mechanism public → fix	Source
2023-01-25	Profile {LINK REMOVE D} & cooldown	Yes	Links in profile text	—	FAQ ↗
2022-10-24	Login overhaul & cookie truncation	No change	Infinite stolen sessions	(IAuthenticationService token migration)	news ↗
2022-10-12	Mobile App 3.0 QR login	Yes	Session deauthorisation panel	—	news ↗
2022-05-10	Trade-cancellation endpoints removed	No change	Offer-swap (cancel & re-send)	≈5 years after the 2017-era scam panels	issue #325 ↗
2020-05-25	Dota 2 7-day trade lock	Yes	Fast item flipping	—	blog ↗
2018-03-29	CS:GO 7-day trade lock	Yes	Instant bot laundering	≈20 months after the July 2016 statement	blog ↗ archive d 2025-02-26 ↗

Sources: the linked Valve posts, help pages and GitHub issues, as dated. Three decisive fixes shipped with no changelog (2022-05, 2022-10, 2023-12) and were detected by users, not announced by Valve — that asymmetry is part of the enforcement record. Lags are computed only where the dossier's own dating (July 2016 statement; 2017-era scam panels) supplies the start point; "—" means no such start point is established here.

[Download PDF](#)

[Download JSON](#)

[Download PGP signature](#)

PhishDestroy · file integrity signature for this fact

[Material record ↗](#)