

Section 5.1

**The standards were public and dated: Valve builds against them every day**

Investigated by: Agent Lucia

The standards were public and dated: Valve builds against them every day

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.**

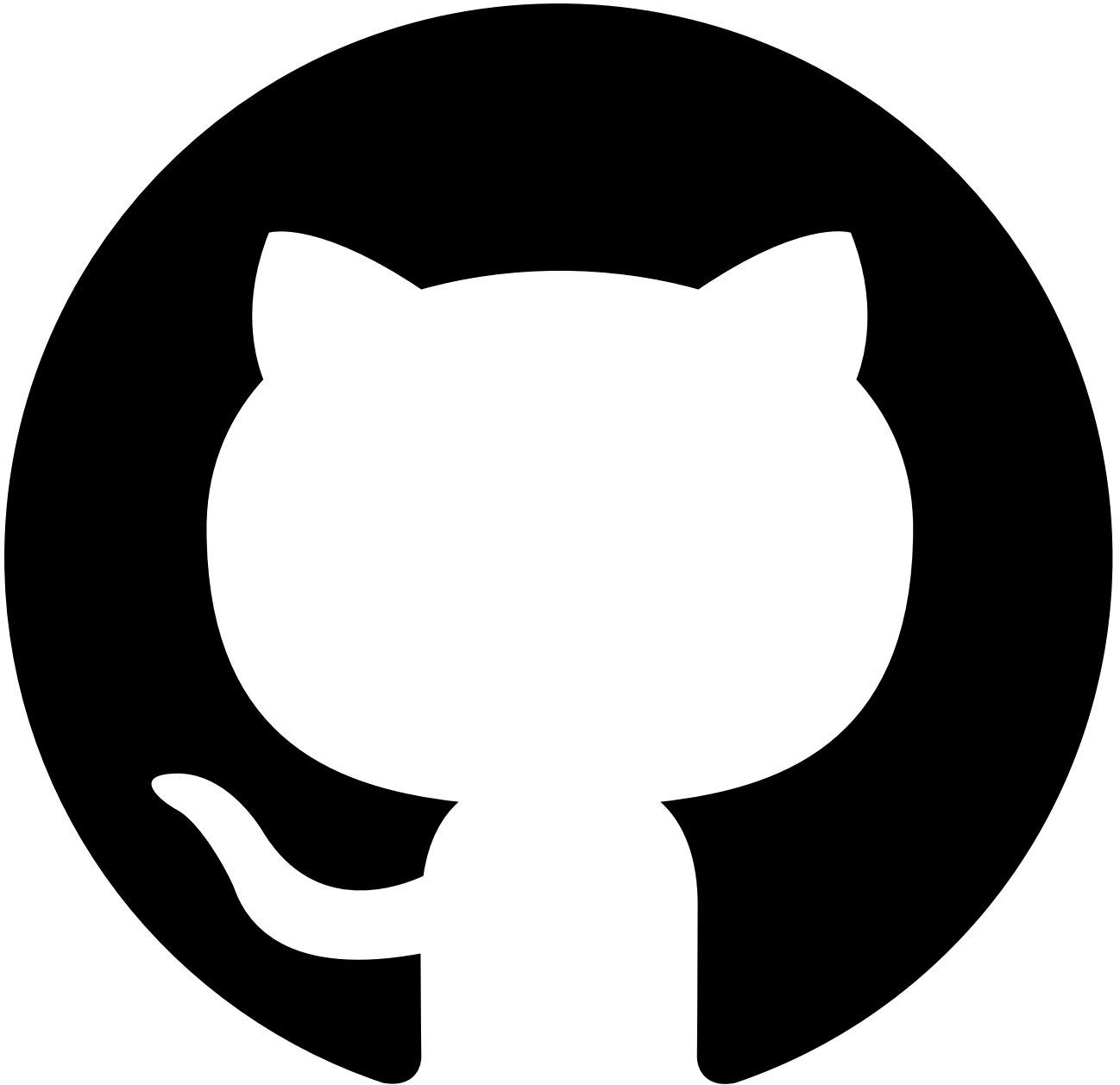
Published external security requirements, with their dates, set against the date the Steam Web API acquired the equivalent control

| The published standard, and the date it took effect                                                                                                                                                                                                                                                                                                                                                                                                          | The Steam Web API on the same question                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Scoped authorisation</b> OAuth 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.                                                                                                                                                                                                                                                       | <b>Granular scopes arrived on 16 January 2026.</b> Before that a single Steam Web API key carried full read and write over the account's trades regardless of why it was issued. <b>Thirteen years and three months</b> after the specification that defines the control.                                                                                                                                                                      |
| <b>A key is not an authorisation</b> Google'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."                                                                                                                                                                                                                                    | <b>The Steam Web API key was exactly that.</b> 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.                                                                                                                                                                                                              |
| <b>Multi-factor authentication on privileged access</b> PCI 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." | <b>A second factor was required to create an API key from 4 December 2023.</b> Until then a stolen session cookie was enough, with no prompt on the account holder's phone and no notification afterwards. <b>Four years and nine months</b> after Apple made Valve itself use a second factor merely to sign in to a developer portal.                                                                                                        |
| <b>Credential lifecycle and revocation</b> 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.                                                                                                                          | <b>No published key-lifecycle record.</b> 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.

[Download PDF](#)

[Download JSON](#)

[Download PGP signature](#)

PhishDestroy · file integrity signature for this fact

[Material record ↗](#)