Privacy Policy: Steam Value Checker

Last updated: November 12, 2026 (Steam Family and saved match corrections added)

Steam Value Checker is a Discord bot that prices a list of games and, if you choose, watches your wishlist and tells you when a game goes on sale. This page explains what it does with your data.

`/games value` stores nothing. Wishlist alerts store the data listed under Storage, and only if you use `/me wishlist`.

What the bot receives

- The game titles you type, or the titles in a file you upload.
- The region, store, sort and `paid` options you choose.
- Your Discord user ID. For `/games value` it is held in memory only, to enforce a short cooldown and let you page through your own results. If you use wishlist alerts it is stored (see Storage).

What the bot does not want

- **Game keys in chat.** Anything that looks like a product key is discarded before any lookup. Please do not post keys. (Giveaways are the one exception: see below.)
- **Passwords, cookies or logins** for Steam, Fanatical or any other service. The bot never asks for them. Wishlist import reads only a **public** Steam wishlist.

What it does with the data

- Game titles (and game IDs for wishlist alerts) are sent to **IsThereAnyDeal** (isthereanydeal.com) to look up prices. Their privacy policy applies to that request. Nothing about you, other than the titles or game IDs, is sent.
- If you use **Sign in through Steam**, you sign in on Steam's own page (steamcommunity.com). Steam tells the bot only your public SteamID64; the bot never sees your Steam password. It stores that ID, that it was verified, and your Discord name and picture so a later Steam sign-in can open your account.
- If a moderator warns, times out, kicks or bans you in a server that uses the bot's moderation commands, the bot saves a case: your Discord ID, the moderator's ID, what was done, the reason they typed and when. Server managers can see their server's cases. `/me mydata delete` removes cases about you; `/me mydata export` includes them.
- If you choose **Linked Roles** on the Steam page, Discord asks you to allow the bot to update your Discord profile connection. The bot sends Discord your Steam name, whether it is verified, how

many games you own and your hours played, and uses the permission once. It keeps no Discord token. You can remove the connection in Discord (Settings, Connections).

- When you run `/me wishlist import`, the bot asks **Steam** (steampowered.com and steamcommunity.com) for the public wishlist of the profile you gave. Steam's own policies apply to that request.
- Results are shown only to you (ephemeral replies) and are returned as an embed and a CSV file in that reply.

Storage

`/games value`: nothing is stored. Prices are cached in memory for about 30 minutes and cleared when the bot restarts.

Wishlist alerts: if you use `/me wishlist`, the bot keeps a small database (SQLite) on the machine that runs it. It contains:

- Your Discord user ID.
- The ID of the server where you last used a wishlist command (so the bot knows which server's alert channel to use).
- Your wishlist: game titles, price-service game IDs, whether each came from Steam or was added by hand, your alert rule for each game, and the last price the bot alerted you about (to avoid repeat alerts).
- Your Steam profile ID (SteamID64), only if you imported a Steam wishlist. It is a public identifier.
- Friends you add with `/me friend`: a nickname, an optional birthday (month and day only, no year) and an optional public wishlist page name. Deleted by `/me mydata delete`.
- Streams: if you link your Twitch, Kick or YouTube channel for automatic go-live posts, the bot saves your Discord ID, the server, the site and the public channel name you gave, and checks that public channel every couple of minutes to see if it is live. `/server stream unlink` (or the Streams page) removes it. Other members of the same server can see your name, channel link and whether you are live on the dashboard's Streamers page. If a server has go-live posts set up, the bot saves, while you are live, your Discord ID, the stream title, game and link you typed, and which post it made, so it can update or gray out the post. It also keeps when it last announced you (to avoid pinging people twice) and, if you make an OBS overlay link, a random token tied to your Discord ID and the server. Ending the stream or resetting the link removes them, and `/me mydata delete` clears the rest.
- Profile card: the gaming names you add (Steam, Xbox, PlayStation and so on), your card color, which stats it shows and any banner or picture links are saved with your account, so the website and

`/me profilecard` draw the same card. They are part of `/me mydata export` and removed by `/me mydata delete`. Your browser loads picture links from their sites (which can see your address). When you run `/me profilecard` or `/me steam value`, or save the card from the website, the bot downloads those pictures once to draw them: only https links to images on public addresses, with a size limit. Nothing is uploaded to anyone else, and the picture is shown only to you unless you choose `public:true`.

- Your settings: region, store filter, delivery choice (DM or channel), whether alerts are paused, a count of failed DM attempts, quiet hours and time zone, weekly digest choice, bundle alert choice, free game alert choice, and whether you opted in to sharing with your server.
- If you use the library: the games you add, what you paid for each (only if you enter it), and one daily snapshot of the totals (number of games, total paid, total MSRP, cheapest price, and how many games are in your backlog, being played or finished) used for the value and backlog charts, `/me library trend` and the leaderboard.
- If you link Steam with `/me steam link`: your public SteamID64 and, for each game you own, its Steam app ID, minutes played, when it was last played and, if the bot has a Steam API key, how many achievements you have unlocked of the total (used to mark a game Finished at 100%). The bot reads only what your public profile shows and never asks for a login. `/me steam unlink` removes the link (and, if you choose, the games it added).
- If you use the FanaticalGrabber extension's send button: a SHA-256 hash of your link code, when it was made and last used, and how many sends. The code itself is never stored. `/me library grabber action:off` deletes it. The extension sends game titles only, never keys.
- If you add people to **Steam Family** (`/me steam family`): each person's public SteamID64, their Steam name and the Steam app IDs of the games their public profile lists, so sale alerts and bundle checks can skip games your family already owns. Removing a person, or deleting your data, erases their list.
- If you save a correction with `/games fixmatch`: the title you typed and the game you picked.
- If you use batch redeem: keys you queue with `/games redeem` are kept only until the batch finishes, you cancel it, or 24 hours pass, then erased. Keys you paste on the dashboard page go from your browser to the extension to Steam and never reach this server. For each settled key the bot keeps a SHA-256 hash and a masked hint (first and last five characters), the result and the game name, so repeats are skipped and `/games redeem history:true` can list them. Clear it from the Batch redeem page.
- If you use `/me steam recap`: the hours added at each Steam sync (the day, the game and the minutes gained), kept for three years so the hours-tracked recap works.

- If you set a backlog challenge: your goal for each month. Progress is worked out from your library and the date you marked a game finished.
- If you turn on a public page: visitors can mark a game as a gift. Only a hash of a random code from their browser is stored, with the game and the time, for 45 days. You are shown a count, never who or which.
- If you wishlist a game that is not out yet: its release date, so you can be told on launch.
- If you use `/server compare`: only members who opted in with `/alerts share` can be compared, and the other person sees nothing. Only game titles and hours are shown.
- If you use the calendar: server events a manager adds (title, notes, time, channel and who added it) are visible to that server's members and removed 30 days after they happen. If you make a calendar link, a SHA-256 hash of it, which servers it covers and when it was last used are stored. The link itself is never stored. `/server calendar feed action:off` deletes it.
- If you post with `/server lfg`: the post (game, time, slots), who joined, and the Discord event ID if one was made. Posts are removed shortly after they close.
- Your alert history: what was sent, when, the price and how (DM or channel). It is deleted after 90 days.
- If you use a server's shared list: the games you suggested and your votes.
- A note of which bundles, giveaways and free games you were already told about, so you are not told twice (deleted after 120 days).

For server admins: the alert channel ID you set with `/manage alertchannel`, and the ID of the admin who set it.

Server sharing is off by default. If you opt in with `/alerts share` (or on the dashboard), other members of the server may see your name in a group alert (\"3 members are watching...\") and on the leaderboard, with your number of games and the percent you saved against MSRP. They never see your wishlist, what you paid or any prices you entered. Opt out at any time.

Server shared list: games added to `/server serverlist` and the votes on them are visible to everyone in that server.

Backups: the bot owner may keep daily copies of the database (the last 7 by default). They hold the same data as above, so deleted data can remain in a backup until it rotates out, up to 7 days.

Owner statistics: the bot owner can see totals (how many members, how many games, how many alerts) but not who watches what.

Your data is kept until you delete it with `/me mydata delete`. Removing the bot from a server clears that server's alert channel. `/me mydata export` gives you a copy of everything stored about you.

User input and results are never written to logs. Error logs contain only technical error messages.

Discord keeps your messages and the bot's replies according to Discord's own policies.

Giveaways

If a server manager runs a giveaway, the bot stores: the giveaway settings, your Discord user ID if you enter, who won each prize and its status, and the prize keys the manager entered. Keys are **encrypted** and are deleted once the winner confirms, when the giveaway is canceled or the key is returned to the host, or automatically after a retention period (delivered keys after 14 days, leftover keys 7 days after the giveaway ends). Keys are sent only to the winner by DM or in a reply only they can see. When a winner confirms, a vouch message naming the winner is posted in the server. Giveaway records are removed 90 days after they finish, and `/me mydata delete` removes your entries.

Error log: the bot keeps recent error and crash messages (with the time) in memory and in a small file next to the database, so the bot owner can read them on the dashboard's Admin page. Tokens and keys are masked. Error messages can occasionally mention a Discord ID or command name.

The web dashboard

If the bot's owner turns on the dashboard, you can sign in with Discord instead of using commands.

- **What Discord shares:** the `identify` and `guilds` scopes, which give the dashboard your Discord ID, name, avatar and the list of servers you are in (with your permissions there, to know where you can set an alert channel). No email, no messages, no passwords.
- **What is kept:** your name, avatar link and server list are held **in memory only** for your session (7 days at most) and are lost when the bot restarts. Discord's access token is used once at sign-in and then discarded. The wishlist and settings you create are stored exactly as described under Storage.
- **Cookies:** one strictly necessary cookie that keeps you signed in, and a short-lived one used during sign-in. There is no tracking or analytics.
- **Alert channels:** when a server admin chooses an alert channel on the dashboard, the bot posts a short confirmation there.

How alerts reach you

- **Direct message by default.** If you use wishlist alerts, the bot will message you without being asked each time. Pause them with `/alerts pause`, or delete your data with `/me mydata delete`.
- **Server channel as a fallback, or by choice.** If a DM cannot be delivered, or you choose channel delivery with `/alerts delivery`, the alert is posted in the alert channel a server admin set. **Everyone who can see that channel can see the game name, and you are mentioned.** If no channel is set, a failed DM is not delivered and the bot tries again on its next check.

Sharing

The bot does not sell, share or transfer your data, other than the titles or game IDs sent to IsThereAnyDeal and the profile lookup sent to Steam, as described above.

Your control

- `/me mydata export` downloads everything stored about you.
- `/me mydata delete` deletes it all.
- `/alerts pause` stops alerts but keeps your list.
- `/games value` stores nothing, so there is nothing to delete for it. You can remove the bot from your server at any time.

Contact

Questions: Jacob Lucado on X.

Changes

If this policy changes, the date above will be updated.

Newer features

- **Public wishlist page:** off by default. When you turn it on, anyone with the link can see your game names, artwork and today's best prices. Nothing else, and never your Discord account. Turn it off any time with `/me publicpage enabled:false`.
- **Backlog status:** the status you set on library games (backlog, playing, finished, dropped) is stored with your library and deleted with `/me mydata delete`.
- **One-off watches:** a game added with `/games watch` is stored like a wishlist game and removed after its alert or after 30 days.
- **Server stats:** numbers for members who opted in with `/alerts share`. Counts and totals only; nobody is named for what they want.
- **Status history:** the bot stores per-day uptime counts and a short log of outages. Nothing about people.

Supporter passes and overlays

- **Supporter passes:** a pass is a one-time payment made on Square's own checkout page. The bot never sees your card or any payment details. Square tells the bot that a payment was made, for how much, and which Discord account the pass is for. The bot keeps that (your Discord ID, the amount, the date and how many days it bought) to run the pass, handle refunds and keep a record of sales. If you

delete your data with `/me mydata delete`, the payment stays as an entry with an amount and a date and nothing that says who you are.

- **Gifts:** if someone gifts you a pass, the bot stores that it was a gift and who it came from long enough to send the thank-you messages. Deleting either person's data clears that link.
- **Founding supporter and the supporter wall:** the first 50 people with a paid pass get a Founding supporter mark. The thank-you wall on the Chip in page only shows a name you type in yourself, and only if you choose to be on it. It is never linked to your Discord account, and you can clear it any time.
- **Overlay studio:** each overlay you make (goal bar, server stats, timer) is stored with its settings and a private link. Anyone who has the link can see that overlay and nothing else, so treat it like a password. Making a new link stops the old one working. Overlays are deleted with your data.
- **Server rivalries:** servers with the weekly digest on are compared once a month using totals only (games finished and hours among members who opted in with `/alerts share`). The other server sees the totals and the server name, never a member's name or library.
- **Wishlist match:** when you look at someone's public wishlist page in Discord, the bot compares it with your own list for that moment and shows the games you both want. Nothing is stored.

Community features (servers that use them)

- **XP and levels:** per server, your Discord ID with your XP, level, a count of counted messages and minutes in voice. Message text is never read or stored; the bot only notes that you sent one. Shown on `/server rank`, `/server levels` and the dashboard to people in that server.
- **Invites and referrals:** which invite you joined with, who it belonged to, and when you joined and left, so inviters can be credited. Shown as counts and names of inviters in that server. Your own invite link is stored if you make one with `/server invites link`.
- **Welcome and leave notes:** the bot posts your name and avatar link in the server's welcome channel when you join, and optionally when you leave.
- **Emoji import:** the bot downloads the emoji or image you ask it to add from Discord's own servers and uploads it to your server. Nothing is kept.
- **Top.gg votes:** if you vote for the bot on Top.gg, your Discord ID, the time of the vote and a running count of your votes are kept so the bonus XP is paid once. `/me mydata delete` removes them.
- **Control:** `/me mydata export` includes this data and `/me mydata delete` removes your XP, joins and links (invites you made stay as counts without your name). If the bot leaves a server, that server's community data is deleted.

- **Cross-server giveaways:** the post shows the host (the bot owner or a partner) and the partner's public links, and optional steps such as following an account. The bot cannot see whether you did them.

Support tickets, notifications and language

- **Tickets:** what you write in a ticket and the replies are stored with your Discord ID and name. They are visible to you, to the bot owner, and, if you picked a server that set up a ticket channel, to that server's staff. Never put a game key in a ticket; staff cannot see giveaway keys. Closed tickets are deleted after 6 months, and `/me mydata delete` removes yours. `/me mydata export` includes them.
- **Notification center:** a copy of the title and short text of each alert sent to you, kept 60 days, only visible to you.
- **Settings:** muted alert types, language and the last "what's new" you saw.
- **Voice channels:** for `/server gamenight` and the optional game night nudges the bot checks who is in a voice channel at that moment. It never records voice, audio or how long anyone was there, and only members who opted in with `/alerts share` are used.

AI companion (only in servers that turn it on)

- Until the full AI is switched on (it is coming soon), the bot answers the common questions itself and sends nothing to anyone. The points below apply once it is on.
- When someone writes "Hey AI ..." or mentions the bot in a server where a manager switched it on, the question, the last few messages of that reply thread and the asker's display name are sent to the AI service to write the answer: **Anthropic** unless the bot owner or the server picked another one (OpenAI, Google Gemini, OpenRouter, Groq, Mistral, xAI or DeepSeek). That service's privacy policy applies to the request. Nothing else from the channel is sent.
- To answer, the AI can call the bot's own lookups: game prices, free games, the asker's own wishlist and library summary, and the server's opt-in savings leaderboard. The results of those lookups for that question are sent along. It only sees the asker's own lists, never another member's, and it can add a game to the asker's wishlist only when they ask.
- A server manager can add that server's own AI API key (for any of those services) in the dashboard. It is stored encrypted, only that server's questions use it, and it can be removed at any time.
- The bot keeps counts only (answers per day, tokens used, per-server daily totals), not the questions or answers.