



# TypVault

Running a tip program: the operator screens, step by step

Screens shown are from a demonstration deployment with sample data. The program shown, Capital City Tip Line, is fictional.

# Contents

|                                           |    |
|-------------------------------------------|----|
| Running a tip program                     | 3  |
| Getting started                           | 6  |
| Signing in and two-factor authentication  | 10 |
| Your Home dashboard and triage            | 14 |
| Working the Workspace queue               | 17 |
| Reading a tip                             | 20 |
| AI insights on a tip                      | 23 |
| Assigning tips                            | 26 |
| Messaging a tipster                       | 29 |
| Entities and entity search                | 33 |
| Routing, referring, and transfers         | 36 |
| Call-center intake from the operator side | 40 |
| Statuses and the case lifecycle           | 43 |
| Rewards                                   | 47 |
| Crime of the Week authoring               | 51 |
| Insights, reports, and export             | 54 |

|                                      |    |
|--------------------------------------|----|
| Agencies, MOUs, and outreach         | 59 |
| Recipient portal administration      | 63 |
| Sensitivity classes and compartments | 66 |
| Access Review                        | 69 |
| Program administration               | 73 |

# Running a tip program

This section covers every task involved in running a tip program on TypVault, from your first sign-in through daily work to program administration.

## Getting started

- [Promises you can keep](#): how the platform turns your program's promise to reporters into a property of the software.
- [Getting started as an operator](#): what's in the console the first time you sign in, and where to start work.
- [Signing in and two-factor authentication](#): signing in with your program's account, setting up your second factor, and what to do if you lose it.
- [Home dashboard and triage](#): what you see when you sign in, and how to decide what to work on first.

## How a tip travels

Seven short articles, in order, following one tip through the whole life of a program:

- [Intake: nothing that identifies the reporter](#)
- [Dialogue: a conversation the platform doesn't track](#)
- [Investigation: give investigators the full picture](#)
- [Evidence: handle evidence safely](#)
- [Referrals: governed at both ends](#)
- [Rewards: pay without names](#)
- [Retention: retain with discipline](#)

## Daily work

- [Working the Workspace queue](#): the queue of tips waiting for you, and how to work through it.
- [Reading a tip: the audited decrypt](#): opening a tip to read it, and why that open is recorded.
- [Statuses and the case lifecycle](#): the stages a tip moves through, from new to closed, and how you move it between them.
- [Sensitivity classes and compartments](#): how sensitivity classes and compartments limit which staff can open a case.
- [Entities and entity search](#): the people, places, and things named across your tips, and how to search them.
- [Messaging a tipster](#): replying to a tipster in a case's message thread, and what they see on their end.
- [AI insights on a tip](#): the automated read on a tip's content, when it runs, and what it can never see.
- [Routing, referring, and cross-program transfers](#): sending a tip to a recipient or team in your program, or referring it to a paired partner program.
- [Assigning tips](#): handing a tip to a specific investigator at your program.
- [Rewards: authorize, dispatch, fund](#): approving a reward, funding it, and getting it to the tipster.
- [Insights, reports, and export](#): the numbers your program can pull, and what an export leaves out.
- [Call-center intake: the operator view](#): what a tip taken by phone looks like once it reaches your queue.
- [Authoring a Crime of the Week](#): publishing a public case for your community to see.

## Administration

- [Access review](#): checking who has an account at your program and what they can reach.
- [Program administration](#): the program-wide settings a program administrator manages.

- [Administering the recipient portal](#): setting up the agencies your program sends tips to, and who can access each one.
- [Agencies, MOUs, and outreach](#): adding a new receiving agency and keeping its paperwork current.

## Related

- [Submitting a tip](#)
- [Taking tips by phone](#)
- [Receiving tips at your agency](#)
- [Transparency](#)

If a task isn't covered here yet, ask your program's administrator.

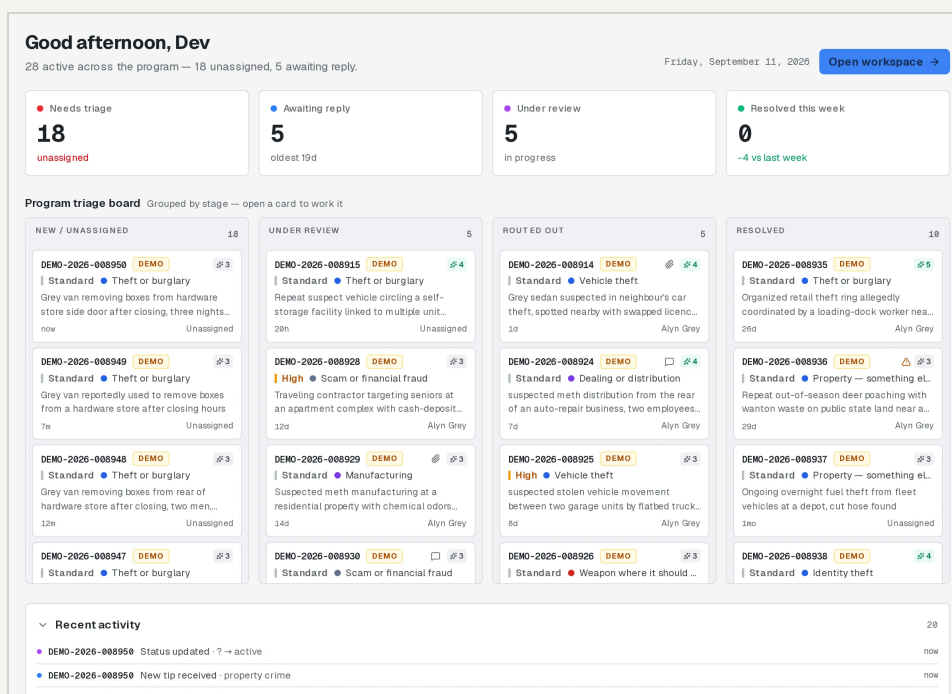
# Getting started as an operator

Sign in and you land on Home. From there, the console has two other places you'll use every day: Workspace, where you open and work a tip, and Insights, where you see how the whole program is doing.

## Home: your queue, at a glance

Home is built around your own work. If you hold a standard operator role, it shows the tips assigned to you: how many need a first look, how many you're waiting to hear back on, and how many you closed this week. If you're a program admin, Home shows the whole program instead.

Below those counts, Home groups your active tips by stage — new, under review, routed out — so you can see where things stand. That grouping is a view: to move a tip forward, open it in Workspace and change its status there.



*Home: your tiles and the triage board.*

# Workspace: where you do the work

Workspace is one screen with three parts: a queue on the left, the tip you've opened in the middle, and a set of actions on the right.

Opening a tip decrypts it so you can read it. That open is attributed to you and timestamped — every tip you look at leaves that record, with no way around it. See [Reading a tip: the audited decrypt](#) for the full picture.

From the action panel, you can set a sub-status or priority, assign the tip, route it to another team, refer it to another program, log a reward, or reopen a closed tip. Each of those is attributed and timestamped the same way opening the tip is.

If the tip has a message thread with the tipster, you'll find it in the same view — see [Messaging a tipster](#) for how that works.

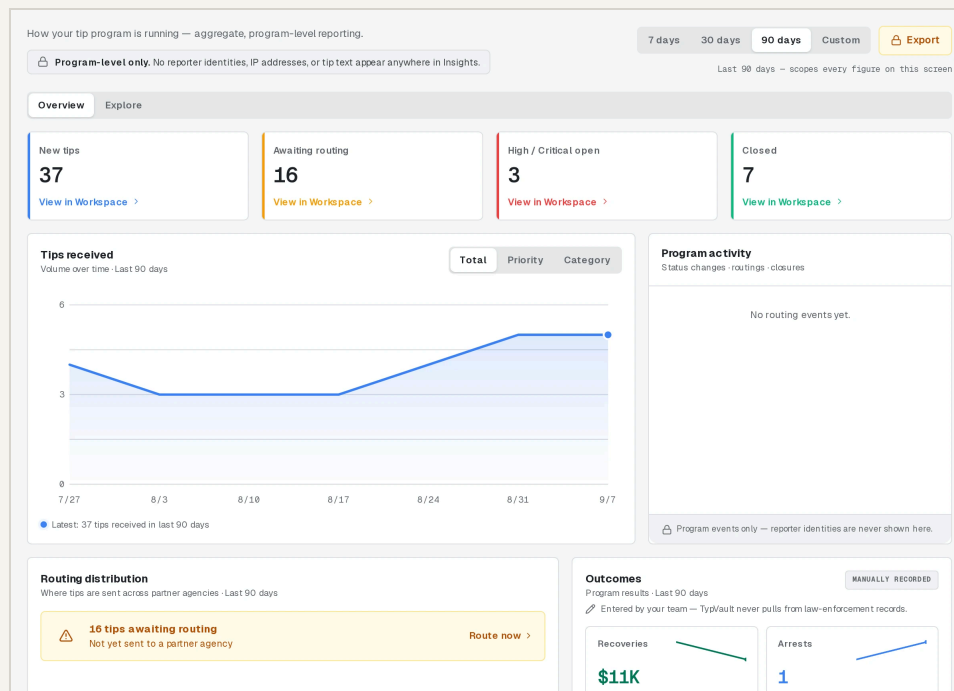
The screenshot displays the 'Workspace' interface, which is divided into three main sections:

- Queue (Left):** A list of tips with filters and status indicators. The queue is titled 'QUEUE 37' and shows 'All 37', 'Active 27', and 'Resolved 10'. It includes a search bar and a 'FILTERS' section. The tips listed are:
  - DEMO-2026-008913: Critical, Dealing or distribution, 4, 10. Status: Active - Pending Assignment.
  - DEMO-2026-008916: High, Theft or burglary, 4, 10. Status: Active - Pending Assignment.
  - DEMO-2026-008925: High, Vehicle theft, 3, 3. Status: Active - Routed.
  - DEMO-2026-008928: High, Scam or financial fraud, 3, 3. Status: Active - Under Review.
  - DEMO-2026-008933: High, Fraud — something else, 2, 10. Status: Active - Actionable.
  - DEMO-2026-008949: Status: 5.
- Tip Detail (Center):** A detailed view of a tip titled 'DEMO-2026-008914'. It shows a progress bar with stages: Received, Routed, Under review, Actionable, and Closed. The tip is categorized as 'Vehicle theft' and was submitted on Sep 10 at 10:27a. It includes an 'AI READOUT' section with a summary: 'Grey sedan suspected in neighbour's car theft, spotted nearby with swapped licence plate'. Below this is a 'NARRATIVE' section with a report from the tipster: 'My neighbour's car was taken from our shared lot on Tuesday night. On Thursday I saw the same grey Honda that has been circling the block for two weeks parked two streets over with a different plate screwed on crooked. The original plate was 7TRV882 — I wrote it down when it kept driving past.' The location is listed as 'FROM 1188 Harborview Rd, Harbor Bay, UT'. There are also buttons for 'Map' and 'Satellite'.
- Actions (Right):** A panel of actions for the selected tip. It includes buttons for 'Set sub-status', 'Set priority', 'Route this tip', 'Refer to another program', 'Log reward', 'Close', 'Close as noise', and 'Pause reporter dialog'. There is also a 'Reassign' button and a 'Routing history' section.

*Workspace: queue, tip, and actions in one place.*

# Insights: the program-level view

Insights doesn't belong to any one tip. It's where you see how tips move through the stages, where cases are piling up, and pull the data you need for reporting. Most days, you'll live in Workspace and Home; Insights is for when you need the wider picture.



*Insights: the program-level view.*

## Where to start

1. Sign in and set up your second factor if you haven't already — it's required before you can reach any case. See [Signing in and two-factor authentication](#).
2. Open Workspace and pick a tip from the queue, so you can see how the three panes fit together.
3. Try one action on that tip — a sub-status or a priority — so you know where the controls live before you need them under pressure.
4. Check Home at the start of each session; it's built to tell you what needs you first.
5. Look at Insights once you've worked a few tips and there's something worth measuring.

Each tip you open also carries a short automated readout, described in [AI insights on a tip](#) — it's a supplement to your own reading of the tip, not a replacement for it.

## Related

- [Signing in and two-factor authentication](#)
- [Reading a tip: the audited decrypt](#)
- [Messaging a tipster](#)
- [AI insights on a tip](#)

If your console layout doesn't match what's described here, your program admin can tell you what's turned on for your role.

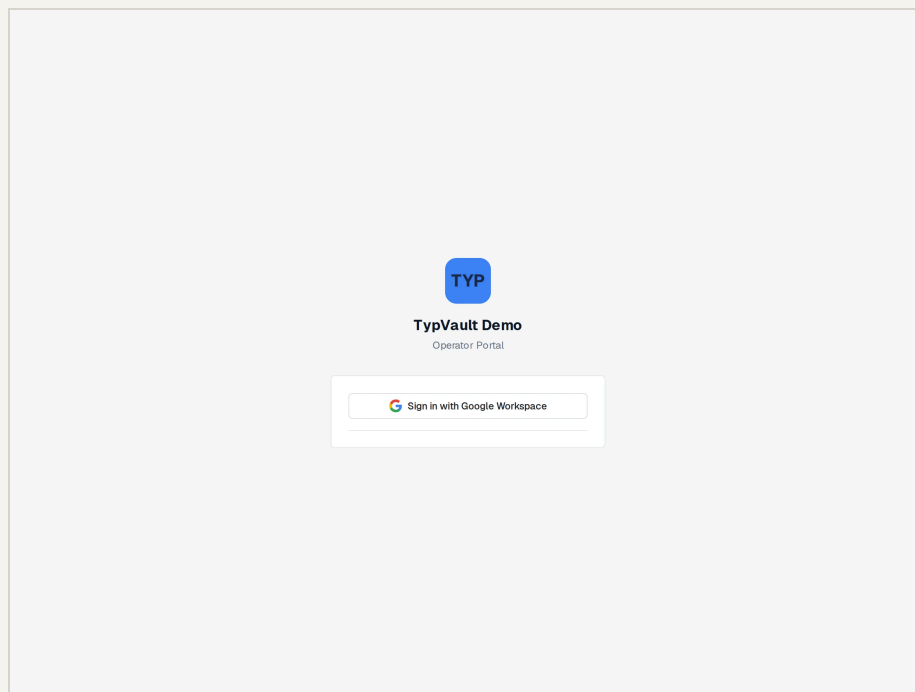
# Signing in and two-factor authentication

You sign in with your program's Google Workspace account, then confirm a second factor. A properly configured deployment requires one for every operator account, with no lighter tier and no grace period after your first sign-in.

## Sign in

1. Open your program's sign-in page.
2. Select **Sign in with Google Workspace** and complete Google's own prompts.
3. TypVault checks your account against the program's operator list. Google confirms who you are; TypVault confirms you belong here.

If your account already has a working second factor, you're asked to verify it next. If not, you're sent to set one up first. It can't be skipped, so budget a couple of minutes the first time through.



*The sign-in screen.*

# Set up your second factor

Enrollment uses an authenticator app: Google Authenticator, 1Password, Authy, or anything that reads a QR code.

1. Open the authenticator app on your phone.
2. Scan the QR code TypVault shows you, or type in the setup key by hand if scanning isn't an option.
3. Enter the 6-digit code the app generates, to confirm the setup worked.
4. Save the ten backup codes shown next. TypVault shows them exactly once: write them down or keep them somewhere safe before you move on.

Each backup code works one time. They're a fallback for when your authenticator app isn't handy, not an everyday sign-in method.

## Note

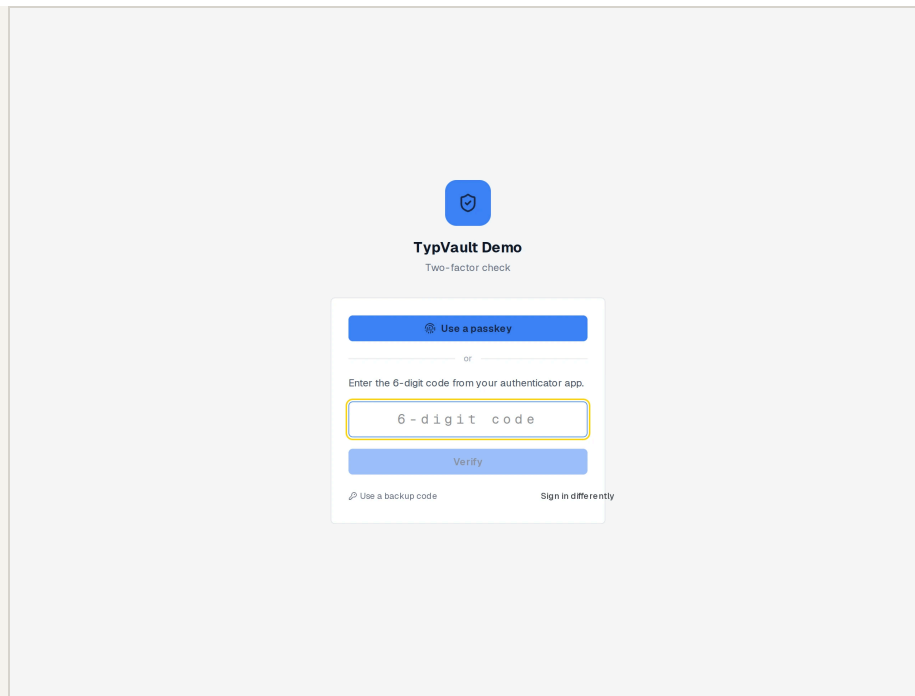
An unfinished setup doesn't count. If you start enrolling and stop partway, restarting issues a fresh QR code and the old one stops working.

# Verify at sign-in

Once you're enrolled, every sign-in asks for your second factor before you reach any case data.

- **With a passkey:** select it and complete your device's own prompt (fingerprint, face, or security key). One tap and you're through.
- **With your authenticator app:** enter the current 6-digit code.
- **With a backup code:** switch to backup-code entry and enter one of your saved codes. Each one works only once.

Too many wrong codes in a row locks the attempt for a short time. Wait it out, or try a different factor if you have one enrolled.



*The two-factor check.*

## Add a passkey

A passkey is additional, not a replacement. You set up your authenticator app first, then add a passkey afterward from your profile page. It's a faster, phishing-resistant way to verify: no code to type, just your device's own fingerprint, face, or security-key prompt.

Add as many passkeys as you use devices. Each one is labeled so you can tell them apart, and you can remove any of your own passkeys yourself at any time. Your authenticator app stays as your base factor, so removing a passkey never locks you out.

## If you lose your authenticator app

Lost your phone or your backup codes? You can't reset your own second factor. That's by design, since a factor reset is a security downgrade that needs a second person to approve. Ask a program admin to reset it for you.

A program admin resets your factor from the user list. Once they do, your old authenticator app and backup codes stop working, and you set up a new one the next

time you sign in. The reset itself is logged.

### **Caution**

A factor reset doesn't end an already-open session by itself. If a device or account may be compromised, ask your program admin to suspend the account too.

## Related

- [Running a tip program](#)
- [Transparency](#)

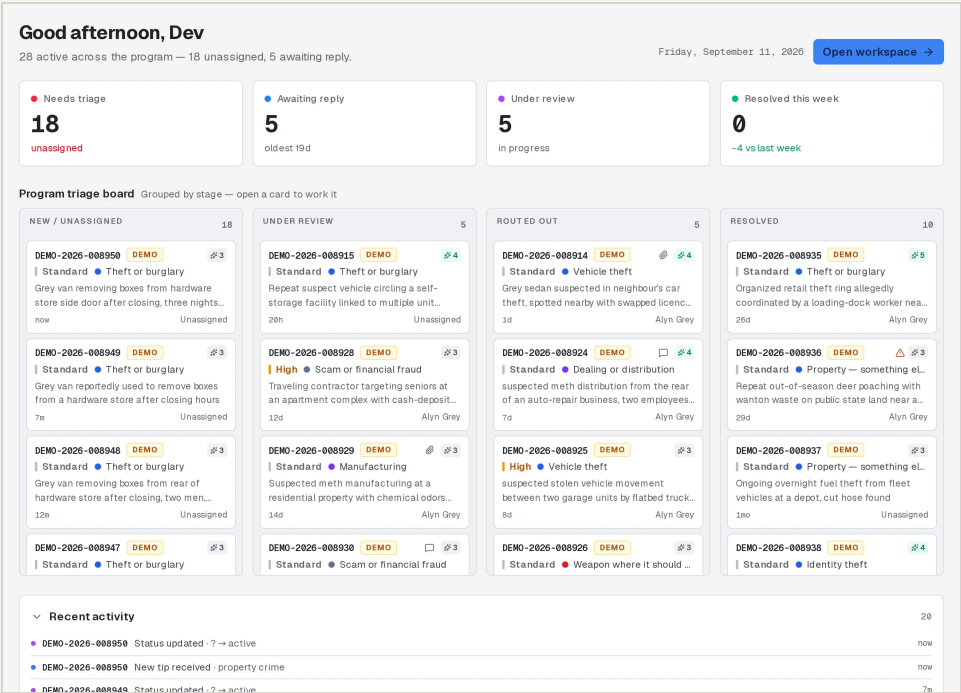
If your sign-in screen looks different from what is described here, ask your program admin.

# Your Home dashboard and triage

Home is the page you land on when you sign in: a personal snapshot of your caseload, not the place where you work a case.

## What you land on

A short line at the top tells you where things stand — how many tips need you, and whether any are critical. Next to it, a button takes you straight into your Workspace: **Go to my queue** for most staff, **Open workspace** for a program admin, since a program admin’s Home covers the whole program rather than one person’s queue.



*What you land on.*

## Your tiles

Four tiles sit below the greeting. For most staff they read: tips that need you now, replies you haven’t answered yet, cases you’re currently reviewing, and how many you closed this week against last week. A program admin sees the same four ideas at the level of

the whole program instead of one queue — how many tips are still unassigned, how many are awaiting a reply, how many are under review, and the week-over-week close count.

You can drag a tile to put whichever one you check first, first. That order, and whether the activity feed further down is expanded or collapsed, follows your account — it's still set the next time you sign in, even from a different computer.

## Reading your triage board

Under the tiles is a board with four columns. It's a live picture of where your tips already stand, built from the same status a tip carries in the Workspace — nothing changes by looking at it. Moving a case forward happens in the Workspace, not here.

- **New for me** (**New / unassigned** for a program admin) — tips that haven't been marked under review or routed out yet.
- **I'm reviewing** (**Under review** for a program admin) — tips currently marked under review.
- **Routed out** — tips you've already routed somewhere else.
- **Resolved** — your most recently closed tips, newest first.

Each card leads with a short case reference — never the tip's internal record ID — plus a priority mark and a category label. A message icon means a reporter has replied since you last looked; a paperclip means the case has an attachment. Some cards also carry a short descriptor line and a small quality score, but only once a tip has already been read by the automated analysis described in [AI insights on a tip](#) — until then, the card just doesn't show one.

Click any card to open the case in your Workspace.

## Reporter replies, your day, and team workload

Below the board, a short list surfaces cases where a reporter has replied or something changed since you last checked, newest first, each linking straight into the Workspace. Standard staff also get a short rundown of the day underneath: what's assigned, what's still critical, and how many you resolved this week. A program admin sees a team workload view in that spot instead, showing how active tips are currently spread across the people on the program.

## Recent activity

Further down, a collapsible feed lists recent events — new tips, status changes, messages, rewards, outreach — limited to whatever you have permission to see, each one linking back to its case when it has one. Collapse it if you don't need it open every time; your choice is remembered.

## Related

- [Getting started as an operator](#)
- [Statuses and case lifecycle](#)
- [Reading a tip: the audited decrypt](#)
- [AI insights on a tip](#)

If Home shows something that doesn't match what you actually have, ask your program admin — role and compartment settings shape what lands here, and your program's exact setup may differ from this page.

# Working the Workspace queue

Every tip your program receives lands in the Workspace queue: one card per tip, in the list on the left of the screen. Select a card and its evidence, the reporter's dialog, and the actions you can take on it load beside the queue, in the same view.

## Filter the queue

The screenshot displays the 'Workspace queue' interface. On the left, a list of tips is shown, each with a status (e.g., 'Active', 'Resolved'), a priority (e.g., 'High', 'Critical'), and a brief description. The selected tip is 'DEMO-2026-008914', which is 'Active' and 'Routed'. The right panel shows the details for this tip, including a 'Safety check ran clean' status, a 'Vehicle theft' category, and a 'Submitted Sep 10' date. The 'AI READOUT' section provides a summary of the tip, stating 'Grey sedan suspected in neighbour's car theft, spotted nearby with swapped licence plate'. The 'NARRATIVE' section contains the reporter's account, and the 'LOCATION' section shows the address '1188 Harborview Rd, Harbor Bay, UT'. The right panel also includes a 'Workflow Actions' section with buttons for 'Set sub-status', 'Set priority', 'Route this tip', 'Refer to another program', 'Log reward', 'Close', 'Close as noise', and 'Pause reporter dialog'. The 'Activity' section shows 'Both', 'Comments', and 'Audit' tabs. The 'Routing History' and 'Status History' sections are also visible.

*The Workspace queue with a tip open.*

Three tabs sit above the list: **All**, **Active**, and **Resolved**. Active means the tip is still open; Resolved covers anything closed, however it got there.

Below the tabs, a search box narrows the list by reference number: your program's own investigator reference, or an external case number if an agency has already supplied one. Press / from anywhere in the queue to jump straight to it.

Cards sort by priority, highest first. Each one also carries a few flags — unread messages, tips linked to this one, attachments — and a card you haven't opened yet, or that's had new activity since you last looked, stands out from the rest.

## Open a tip

Click a card to open it. Or move a highlight through the list with **J** and **K** and press **Enter** when you're on the one you want. The highlight moves on its own, so you can scan the queue before committing to a tip.

Opening a tip decrypts it and writes a row to its audit trail: that's a disclosure event, not a routine page load. See [Reading a tip: the audited decrypt](#) for what gets recorded and why it can't be undone.

## Change what happens to a tip

Once a tip is open, its actions sit in the panel next to it. What's available depends on where the tip already is and what your role allows:

1. **Assign or reassign** — pick an operator from a searchable list, and set the agency's own case number alongside it if one exists.
2. **Set sub-status** — moves the tip along its status steps: Received, Routed, Under review, Actionable. The options shown depend on where it already is, and this turns off once the tip is closed.
3. **Set priority** — critical, high, standard, low, or lowest.
4. **Route this tip, or refer it to another program** — sends it to a recipient on your own team, or to a partner program your admin has paired with. See [Routing, referring, and cross-program transfers](#) for how each works and what the other side sees.
5. **Log a reward** — record the amount, the reason, and, if you have them, the outcome type and a case reference.
6. **Close** — pick a closure outcome, a sub-status under it, and a reason.
7. **Close as noise** — for a tip that's out of scope, a duplicate, or otherwise not worth investigating. That's a separate call from a closure outcome, and it asks you to confirm before it goes through.
8. **Reopen** — available once a tip is closed, within your program's reopen window. A reason is required, and past that window, only a system administrator can bring it back.

Every one of these writes to the tip's audit trail the moment it happens, with your name attached. There's no quiet version of any of them.

## Related

- [Getting started as an operator](#)
- [Reading a tip: the audited decrypt](#)
- [Routing, referring, and cross-program transfers](#)
- [AI insights on a tip](#)

If an action is missing or grayed out for your role, your program admin can tell you what's turned on for it.

# Reading a tip: the audited decrypt

A tip's content stays sealed until an operator opens it. The moment you open it, TypVault decrypts it for you. That open is a disclosure event: it gets a row in the audit log, and that row cannot be changed or removed.

This is not a warning. It is how the system is built to work, and it is the same posture whether you are looking at your first tip of the day or your fiftieth.

## What counts as an open

Each of the following decrypts content on your behalf, and each is logged the same way:

- Opening a tip's narrative and details.
- Opening its message thread with the reporter.
- Viewing its location on the map.
- Printing it.
- Running an AI read on it, or a similarity check against other tips.

If a tip moves to or from another program, the decrypt needed to prepare that move is logged the same way, too.

## What the log records

Every disclosure row holds three things:

- **Who:** the operator who opened it.
- **When:** the exact time of the open.
- **Why:** which of the actions above triggered it.

The row is written first, before the decrypt is even attempted. A failed open still leaves a trace.

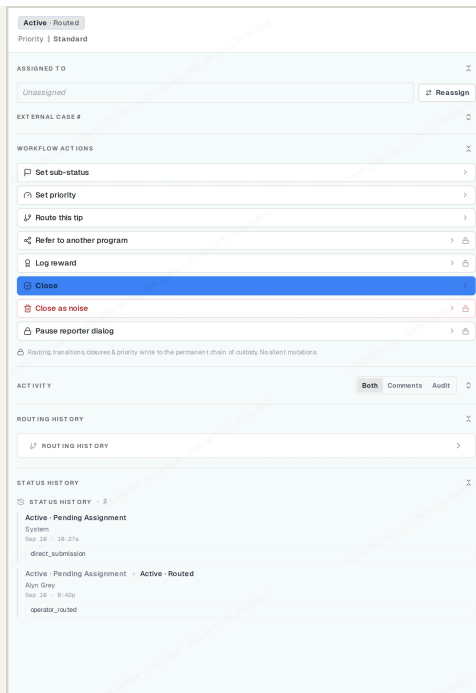
### **Note**

A tip's audit trail defaults to the actions your team took on it — status changes, routing, assignment, notes — not every open. Opens are recorded the moment they happen; the default view just keeps them out of the way so a heavily-reviewed tip's history stays readable. A toggle brings every open into view, and the trail tells you how many are hidden.

## Viewing a tip's audit trail

1. Open the tip.
2. Open its audit view.
3. You'll see the deliberate actions taken on the case, newest first.
4. Turn on the option to show every open if you need the full disclosure history, not just the workflow actions.
5. If a tip has more history than fits on screen, the trail says so rather than showing a partial list as if it were complete.

Some roles do not carry access to a tip's full audit view. If yours doesn't, your program admin can check your access.



*The Audit tab: who opened the tip, and when.*

## Why the log cannot be edited

No operator role, including the highest one in your program, can update or delete a row once it is written. That is enforced below the application, not by a screen that has no edit button. If a tip is ever opened by someone who should not have opened it, the record of that open still exists, unedited, for whoever reviews it later.

## Related

- [Getting started as an operator](#)
- [Messaging a tipster](#)
- [AI insights on a tip](#)

Reporters are told about this same guarantee, in their own words, on the [Operator accountability](#) page.

# AI insights on a tip

When you open a tip in your Workspace, TypVault runs an automated read on its content and shows you the result in the tip's detail view: a quality score, a category check, and a few suggested next steps.

## What the readout shows

- A one-line gist of what the tip is about, plain enough to show safely in a shared queue view.
- A quality score from 1 to 5, with one sentence explaining why.
- A category check: whether the wording agrees, disagrees, or is unclear about the category the tip was filed under, with a suggested category when it disagrees.
- A short list of follow-up questions you could ask the source.
- When the message thread shows the source is worried about being identified, one or two short reassurance lines you could send.

Two more notes appear only when they apply. One flags that the tip might be the source describing their own situation, not someone else's. The other flags that the source may have said something, in their own words, that could point back to who they are. Both are cues for you to weigh, never a determination the system has already made on its own.

QUEUE 37

All 37

Active 27

Resolved 10

FILTERS

DEMO-2026-008913

DEMO

Critical

Dealing or distribution

4

0

6h

Active - Pending Assignment

DEMO-2026-008916

DEMO

High

Theft or burglary

4

0

3h

Active - Pending Assignment

DEMO-2026-008925

DEMO

High

Vehicle theft

3

0

8d

Active - Routed

DEMO-2026-008928

DEMO

High

Scam or financial fraud

3

0

12d

Active - Under Review

DEMO-2026-008933

DEMO

High

Fraud — something else

2

0

20d

Active - Actionable

DEMO-2026-008949

DEMO

0

5

DEMO-2026-008914

DEMO

Unrestricted

Safety check ran clean

Scroll

Tabs

Print / PDF

Vehicle theft

Submitted Sep 10

10:27a

Age 1d

Received

4

Under review

3

Actionable

4

Closed

5

Overview

Dialog

Evidence

Outcomes

AI READOUT

Re-run

Grey sedan suspected in neighbour's car theft, spotted nearby with swapped licence plate

Source may have identified themselves. The source indicates they share a parking lot with the victim and that the reported location is their own neighbourhood, which could narrow their identity to so Handle with extra care.

Category confirmed

QUALITY 4 / 5

2026-09-11 15:02:13

NARRATIVE

REPORTER'S ACCOUNT - AS SUBMITTED

My neighbour's car was taken from our shared lot on Tuesday night. On Thursday I saw the same grey Honda that has been circling the block for two weeks parked two streets over with a different plate screwed on crooked. The original plate was 7TRV882 — I wrote it down when it kept driving past.

Decrypted under your operator key. Opening this tip was recorded as an audited disclosure — the act of reading is the forensic event.

LOCATION

FROM 1188 Harborview Rd, Harbor Bay, UT

Map

Satellite

Active - Routed

Priority | Standard

ASSIGNED TO

Unassigned

Reassign

EXTERNAL CASE #

WORKFLOW ACTIONS

Set sub-status

Set priority

Route this tip

Refer to another program

Log reward

Close

Close as noise

Pause reporter dialog

Routing, transitions, closures & priority write to the permanent chain of custody. No silent mutations.

ACTIVITY

Both

Comments

Audit

ROUTING HISTORY

ROUTING HISTORY

STATUS HISTORY

STATUS HISTORY · 2

*The readout sits at the top of the tip.*

## How it runs

TypVault runs this analysis on its own, as part of opening the tip. You don't have to ask for it. If it hasn't finished yet, or you want a fresh read after new information comes in, an explicit Re-run (or Run Analysis, the first time) repeats it.

### ⚠ Not every tip gets a readout

A tip moved into a restricted-access compartment does not get this analysis, and Re-run has nothing to return there. A tip with no narrative, no follow-up answers, and no subject details has nothing to analyze either. The panel says so plainly instead of showing an empty score.

## No identity to draw on

The analysis works from the tip's words, its category, any structured details that came in with it, and your recent messages with the source. It never draws on anything that could

identify who sent it. TypVault does not ask a tipster for a name or contact information in the first place, so there is no identity for this analysis, or anything else, to reach for.

## Using what it suggests

The follow-up questions and reassurance lines are a starting point, not an instruction. Clicking one drafts it into your message composer. It does not send until you press Send yourself. The readout can point you at a good next question. It cannot decide the case, and it never replaces your own read of the tip.

## Related

- [Running a tip program](#)
- [The safety check](#)
- [Operator accountability](#)

If the readout in your Workspace looks different from what's described here, ask your program admin. The version your program runs may not match this page yet.

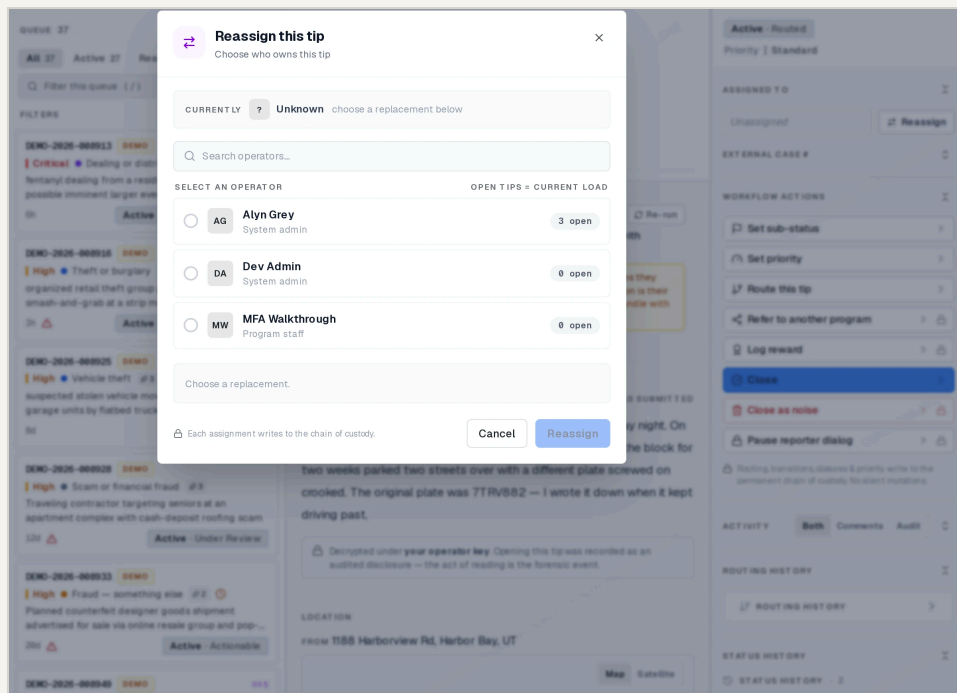
# Assigning tips

Assigning a tip puts one operator's name on it as the owner. You can assign a single tip from its detail view, or assign a batch of tips at once from the queue.

## Assign one tip

1. Open the tip.
2. In the assignment panel, select **Assign**, or **Reassign** if someone already owns it.
3. Search the operator list, or scroll it. Each operator's current open caseload shows next to their name, so you can see who has room.
4. Select an operator.
5. Check the summary line, then confirm.

The operator list only shows active accounts on your program. If you're reassigning, the current owner is marked and grayed out unless you pick them again.



*Assigning one tip from its Assigned to section.*

## Assign a batch of tips

1. In the queue, select the tips you want to hand off.
2. Select **Assign** in the bar that appears above the list.
3. Choose how to split the work:
  - **Assign all to one** puts every selected tip on a single operator.
  - **Distribute round-robin** splits them across two or more operators you pick, weighted toward whoever has the lightest caseload.
4. Review the split before you confirm. It shows exactly how many tips each operator is about to receive.
5. Confirm.

### Note

A batch tops out at 500 tips. If one tip in the batch fails, for instance because your access to it changed after you selected it, the rest still go through. You'll see which failed, and why.

## Who can assign, and to whom

Assigning a tip needs the same access to it as everything else you do with that tip, plus your role's permission to reassign. Your program admin sets that permission up.

You can only assign to another active operator account on your program, not an outside address or an agency contact.

## What this leaves behind

Every assignment and reassignment writes a row to the tip's audit trail: who it went to, who made the change, and when. That row can't be edited or removed afterward, the same as any other entry on the trail.

A separate field next to assignment holds your own agency's case number, if you use one. It's saved on its own and isn't part of the assignment.

## Related

- [Reading a tip](#)
- [Sensitivity classes and compartments](#)
- [Routing, referring, and cross-program transfers](#)

The same guarantee, in plain terms, is on the [operator accountability](#) page.

# Messaging a tipster

You reply to a tipster inside the message thread attached to their case. There is no separate messaging tool, and no address to reach them outside that one thread.

## Find the thread

Two ways to get there:

- **Messages**, which lists every case with reporter activity across your queue. One view shows cases with a reply you haven't opened yet; a second shows every case with a message on either side.
- **The case itself**, open to its dialogue section. It's the same thread, reached from inside the case you already have open.

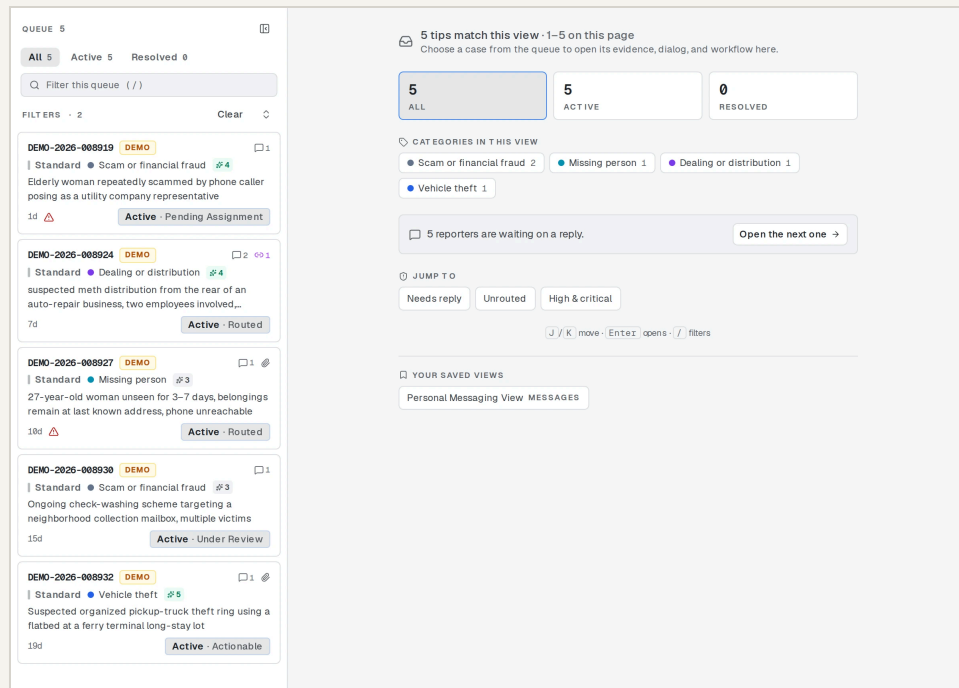
The screenshot displays a software interface for managing cases. On the left, a 'QUEUE' sidebar lists several cases with their status (e.g., Active, Pending Assignment, Routed, Under Review, Actionable). The main area shows a detailed view of case 'DEMO-2026-008935'. This view includes a 'Dialog' tab with a message thread between an 'OPERATOR' and a 'TIPSTER'. The 'OPERATOR' message states: 'An arrest has been made in this case. Thank you for coming forward.' The 'TIPSTER' message responds: 'thank you for telling me. i was worried nothing came of it'. Below the messages, there are 'AI FOLLOW-UPS' and a 'Send' button. On the right, a sidebar contains various case details and actions, including 'ASSIGNED TO' (Unassigned), 'EXTERNAL CASE #', 'WORKFLOW ACTIONS' (Set sub-status, Set priority, Route this tip, Refer to another program, Log reward, Reopen, Pause reporter dialog), 'ACTIVITY' (Both, Comments, Audit), 'ROUTING HISTORY', and 'STATUS HISTORY'.

*The Dialog tab: the thread with the reporter.*

## Reply

1. Open the thread. This decrypts it on the spot, and that open is logged as an access, the same accountability that applies to opening the rest of the case.
2. Read what's there.
3. Type your reply and send it.

The thread carries no identifying signal about the tipster — no name, no device, no location — beyond whatever they choose to write to you.



*Messages: every thread in one place.*

## What reaches them, and when

Your reply does not push out to the tipster. There is no email, no text message, and no notification of any kind. They see it only when they come back and check their case with their own tip code.

Messaging also depends on that case having a passphrase: the extra lock a tipster can add, and the same key that unlocks the thread on their side. If they haven't set one, they can still confirm their case was received, but they can't open any message, including ones you've already sent, until they add a passphrase of their own.

### **Note**

If a case has no passphrase, your reply is written but not readable to the tipster until they set one. Nothing on your side flags this: the composer looks the same either way.

## A limit on how fast you can send

Sending on a single case is capped over a short trailing window. It exists to catch a stuck retry or a copy-paste loop, not to slow down ordinary back-and-forth: a handful of replies in a normal conversation never comes close to it. If you do hit it, wait a moment and send again. It only counts sends on that one case, and your program admin can raise or lower the number it allows.

## When the thread is paused

Two kinds of case can't be messaged:

- **A draft.** The tipster hasn't finished submitting yet, so there's no case to reply to.
- **A closed case.** Closing pauses the thread. The one exception: if a reward on that case is still waiting to be picked up, the thread stays open so you can coordinate the pickup.

Reopening the case lifts the pause for any other reason you need to message again.

## Related

- [Running a tip program](#)
- [Operator accountability](#)
- [Anonymity](#)

If a thread looks frozen and you're not sure why, or a case needs an exception to the send limit, your program admin can check the setting behind it.

# Entities and entity search

An entity is a suspect, a victim, or a vehicle that an investigator has typed into a tip as its own structured entry — separate from the narrative the tipster wrote. Recording one makes it something the console can match against every other tip you're authorized to see, not just the one it came from.

## Suspects, victims, and vehicles

No entity gets created on its own — an investigator decides what belongs in one, reads the tip, and types in the specific facts worth tracking. Three kinds are available:

- **Suspect** or **victim** — a name, aliases, a description, an approximate age, and a phone number.
- **Vehicle** — make, model, color, year, license plate, and plate state.

Every kind also takes a short free-text note. You don't have to fill in every field, but at least one identifying detail beyond that note needs to be there — an entity with nothing but a note won't save.

## Adding an entity to a tip

1. Open the tip and find the **Extracted entities** panel among its details.
2. Select **Add entity**, then choose Suspect, Victim, or Vehicle.
3. Fill in whatever details you have. Read the tipster's account first, then translate the concrete facts — a name, a plate, a description — into the matching fields, rather than pasting in the sentence itself.
4. Save. The entry joins a running, editable list on the tip, grouped by kind.

Any operator who can open the tip can see its entities. Recording, editing, or removing one takes more than a signed-in session — your account needs the base operator role, the same floor that gates most other tip actions.

The kind is fixed once you save an entity; every other field stays editable afterward, and removing one deletes it outright, with no separate archive step. Each add, edit, or removal writes its own entry to the tip's activity trail — the kind of entity and, for an edit, which fields changed, never the values themselves.

## Searching across every tip

Entity Search, in the console's main navigation, runs one query over every entity on every tip you're authorized to see, whether or not it's assigned to you. Pick a type (All, Suspects, Victims, or Vehicles), or switch to the dedicated **License plate** mode, enter your term, and results come back grouped: one result per matching entity, with every tip that mentions it listed underneath.

### Plates need their own mode

The general search box does not match license plates — typing a plate there returns nothing. Switch to License plate mode instead; it also ignores spaces and punctuation, so “K7T 441” and “K7T-441” match the same plate.

A search needs a real term — picking a type alone, with the box empty, returns nothing. Matches are capped at 100; if you hit the cap, narrow it by type, add more of the name or plate, or switch into License plate mode rather than trying to page further.

Your last several searches are remembered on your own operator account and offered back the next time you open the page. They don't show up for any other operator, and they follow you if you sign in from a different device.

Running a search writes one entry to the audit trail — the term you searched, the mode, and how many matches came back, attributed to your account. Opening a tip from the results list is recorded on that tip's own activity trail, the same as any other open.

Search a suspect, victim, vehicle, or license plate across every tip in the program, including tips you were never assigned.

Search

License plate

All types

Q e.g. a name, a phone number, a vehicle description

Q Search

 Searches here are recorded on the audit trail, and opening a tip is recorded in that tip's activity log.



**Start with a name, a vehicle, or a plate**

Type an entity above. This matches it against the structured facts operators have recorded on every tip and groups the results by what matched.

*Entity Search.*

## Related

- [AI insights on a tip](#)
- [Opening a tip](#)
- [Operator accountability](#)
- [Running a tip program](#)

If the Extracted entities panel or Entity Search look different from what's described here, ask your program admin — the version your program runs may not match this page yet.

# Routing, referring, and cross-program transfers

You route a tip to a recipient or team from the tip's own workflow actions. Referring or transferring it to a different program uses the same panel, but only reaches a partner program your own admin has already paired with.

## Route a tip within your program

1. Open the tip and choose **Route this tip**.
2. Pick a recipient (a specific agency or unit in your directory) or a recipient group, which reaches every member at once.
3. Add a note. It's required, and it travels with the routing record for whoever picks up the case next.
4. Send it.

Routing is open to any operator on your team. It doesn't need admin rights.

The screenshot displays a software interface for managing tips. On the left, a 'QUEUE' sidebar lists several tips with their status (e.g., 'Active', 'Resolved') and filters. The main area shows a detailed view of a tip titled 'DEMO-2026-008914'. It includes a progress bar with stages: Received, Routed, Under review, Actionable, and Closed. Below this is an 'AI READOUT' section with a summary of the tip's content. The 'NARRATIVE' section contains a detailed report from the 'REPORTER'S ACCOUNT'. On the right, a 'ROUTE THIS TIP' panel offers options to assign the tip to a specific recipient or group, with a search bar and a list of available recipients. At the bottom of this panel, there are buttons for 'Route', 'Cancel', 'Refer to another program', and 'Log reward'.

*Route this tip: pick the recipient, write the note.*

## What the recipient sees

Routing a tip does not send anything on its own. There is no email, text, or outside alert. TypVault queues a pending notification instead.

- A recipient with a portal account can sign in and see the case themselves, whenever they check.
- Otherwise, someone on your team notifies them the way your program normally does, then marks that notification as sent from the pending list so the record matches reality.

### Note

The pending-notification badge is a queue to work, not proof a message went out. Nothing leaves TypVault on its own when you route a case.

## Refer or transfer a tip to another program

This only works with a paired partner program. Your program admin sets up that pairing once, in settings, before any tip can move. If your program hasn't paired with anyone yet, there's nothing to refer to.

1. Open the tip and choose **Refer to another program**. This step needs program admin rights.
2. Pick an active partner from the list.
3. Choose **referral** (both programs keep working the case) or **transfer** (the destination takes it over).
4. Add a note for the destination. It's required.
5. Send it.

The destination's own operator has to accept before any of the case's content crosses. A short summary reaches them first: category, rough date, how many files. The narrative,

location detail, and files themselves wait for their acceptance. Nobody on the sending side can push a transfer through alone: the destination decides too.

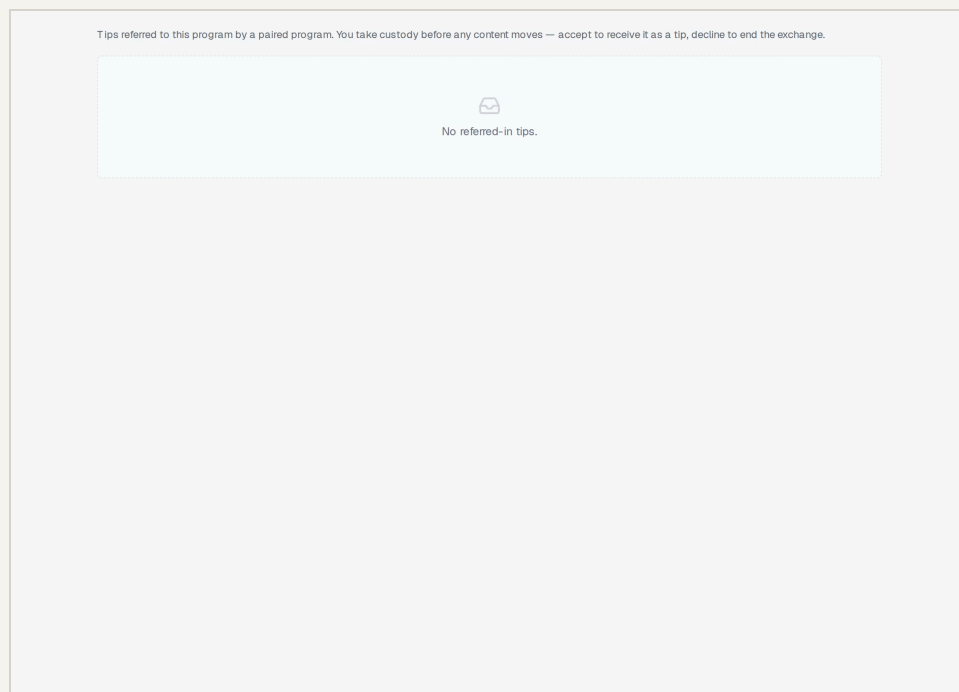
The reporter doesn't choose whether their tip is referred. That decision belongs to the two programs, not to them, and there's no way for them to block it once it's underway.

Once accepted, the case content is re-encrypted just for the destination's own key. Only their investigators can open what arrives. The reporter's tip code, passphrase, and message thread never travel with it. The destination gets the case; it never gets a way into your dialog with the person who reported it.

The reporter's own case view shows that a referral happened, plainly, naming the partner program when it's known. That disclosure isn't something you can turn off case by case.

### **Caution**

Closing a referred or transferred tip starts your own retention clock. When it runs out, the reporter's dialog and reward channel here are removed for good, even if the partner program is still working the case. Use **Closed After Referral**, and close it only once the outcome is on file or by a deliberate call to do so.



*Federation: transfers between programs.*

## Related

- [Running a tip program](#)
- [Referrals to other programs](#)
- [Statuses and case lifecycle](#)

If a pairing needs to be set up, changed, or revoked, that's your program admin's call, in settings, not something you do from a tip.

# Call-center intake from the operator side

A tip taken by phone lands in your queue looking like any other case: the same category, the same structured fields for a suspect, a vehicle, or a victim, and the same free-text narrative. The call-center console builds it from the same pipeline your program's self-service submission uses. A call-taker fills in what the caller says, and the console encrypts it on the spot, before anything reaches TypVault.

The call-taker who captured it worked as a conduit for what the caller said, not as an investigator on it. Once the case is in your queue, nothing about its shape reveals that it started as a phone call. The one visible difference is how the reporter checks back in, covered below.

## What the call doesn't carry

The call itself isn't part of the record. The console keeps no phone number, no caller-side IP address, no location for the caller, and no recording of the call: none of that has a field to land in, on this console or anywhere after it.

A working notepad sits beside the capture form, for anything the call-taker needs to jot mid-call. It's scratch for that one call: it clears the moment the call ends, and it never leaves the call-taker's own screen.

What is on the record is which of your operators captured the case: the same attribution your program already keeps for every action taken on a tip. That entry is about your call-taker's own work. It is never a trace of who called.

QUEUE 37

All 37

Active 27

Resolved 10

FILTERS

DEMO-2026-008913

DEMO

Critical

Dealing or distribution

4

fh

Active - Pending Assignment

DEMO-2026-008916

DEMO

High

Theft or burglary

4

3h

Active - Pending Assignment

DEMO-2026-008925

DEMO

High

Vehicle theft

3

8d

Active - Routed

DEMO-2026-008928

DEMO

High

Scam or financial fraud

3

12d

Active - Under Review

DEMO-2026-008933

DEMO

High

Fraud — something else

2

20d

Active - Actionable

DEMO-2026-008949

DEMO

5

DEMO-2026-008914

DEMO

Unrestricted

Safety check ran clean

Scroll

Print / PDF

Vehicle theft

Submitted Sep 10

10:27a

Age 1d

Received

Routed

Under review

Actionable

Closed

Overview

Dialog

Evidence

Outcomes

AI READOUT

Re-run

Grey sedan suspected in neighbour's car theft, spotted nearby with swapped licence plate

Source may have identified themselves. The source indicates they share a parking lot with the victim and that the reported location is their own neighbourhood, which could narrow their identity to so Handle with extra care.

Category confirmed

QUALITY 4 / 5

2026-09-11 15:02:13

NARRATIVE

REPORTER'S ACCOUNT - AS SUBMITTED

My neighbour's car was taken from our shared lot on Tuesday night. On Thursday I saw the same grey Honda that has been circling the block for two weeks parked two streets over with a different plate screwed on crooked. The original plate was 7TRV882 — I wrote it down when it kept driving past.

Decrypted under your operator key. Opening this tip was recorded as an audited disclosure — the act of reading is the forensic event.

LOCATION

FROM 1185 Harborview Rd, Harbor Bay, UT

Map

Satellite

Active - Routed

Priority | Standard

ASSIGNED TO

Unassigned

Reassign

EXTERNAL CASE #

WORKFLOW ACTIONS

Set sub-status

Set priority

Route this tip

Refer to another program

Log reward

Close

Close as noise

Pause reporter dialog

ROUTING HISTORY

ROUTING HISTORY

STATUS HISTORY

STATUS HISTORY · 2

*A phoned-in tip in the Workspace.*

## Retrieval without a passphrase

There's no passphrase step on a tip taken by phone. Before the call ends, the console generates a retrieval code, and the call-taker reads it aloud. That code alone is what the caller uses afterward to check the case.

That has the same consequence as any case whose reporter hasn't set a passphrase yet. They can check status with the code alone, but the message thread stays locked until they come back and set a passphrase themselves. Your reply, if you've already sent one, is sitting there. It just isn't readable to them until then. See [Messaging a tipster](#) for how that plays out on a case you're replying to.

## The intake check still runs

Every tip gets the same automated check the moment it's logged, phone or not. What differs is which step confirms the content is right. A self-service reporter walks through an on-screen review before they submit. On a tip taken by phone, that confirmation is the

read-back the call-taker does out loud, before the case is logged at all. [That confirmation step](#) is covered from the call-taker's side.

## Where “phone” is visible to you

The case itself carries no separate label for how it arrived, in your queue or in its detail view. The one place the channel shows up on its own is your program's Insights, where tips taken by phone can be grouped and counted alongside every other channel your program takes tips through.

### Note

If you need to know how many tips came in by phone this month, Insights is where to look, not a filter on the queue itself.

## Related

- [Taking tips by phone](#)
- [Messaging a tipster](#)
- [Reading a tip](#)
- [Insights, reports, and export](#)

If a case in your queue was taken by phone and something here doesn't match what you see, check with your program admin. Your program's console may be running a different version of this flow.

# Statuses, sub-statuses, and the case lifecycle

Every tip carries two linked fields: a primary status for the broad stage of the case, and a sub-status for the specific point within it. This page maps both and what moving between them changes.

## The five primary statuses

A tip sits in one of five primary statuses: Draft, Active, Resolved, Unresolved, or Closed. Draft is a submission still in progress on the reporting side. It never reaches an operator queue, so it never appears in your working set.

Active is the normal working state for an open case. Resolved, Unresolved, and Closed are the three outcome categories a case can end in. All three behave as closed for workflow purposes, even though only one of them is named “Closed.”

## Sub-status tracks progress within Active

While a tip is Active, its sub-status shows as a step in the case pipeline: Received, Routed, Under review, Actionable. Received covers both a tip with no sub-status yet and the taxonomy’s own label “Pending assignment.” Both show as the same first step.

Setting a sub-status moves the case forward without changing its primary status. The control turns off once the case is closed.

## Closing a case

Closing a case asks for three things:

1. Which outcome category it belongs in: Resolved, Unresolved, or Closed.

2. Which specific reason within that category, for example Arrest made, Unfounded, or Duplicate.
3. A written reason.

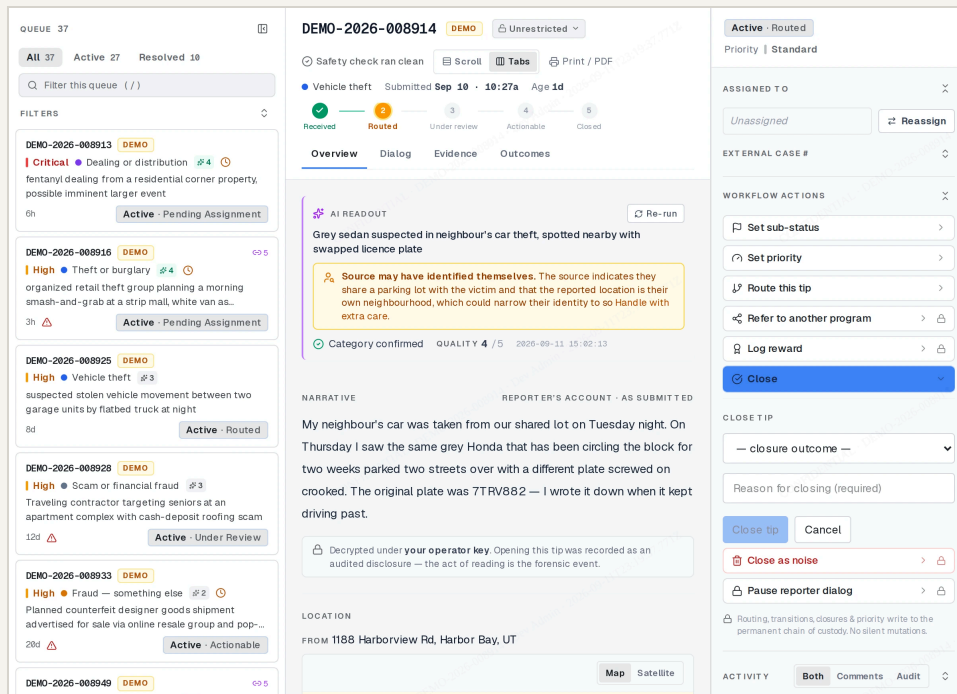
All three are required before the case actually closes.

Closing also freezes the reporter's side of the case. Once a tip is closed, the person who sent it cannot send a new message or attachment through the dialog.

## Note

A reward that is authorized or already issued keeps the dialog open, so a reporter can still arrange collection after the case closes.

Routing and setting sub-status turn off on your side too, once a case is closed.



The screenshot displays a case management interface. On the left, a 'QUEUE 37' sidebar lists several cases with filters for 'All 37', 'Active 27', and 'Resolved 18'. The main panel shows a detailed view of case 'DEMO-2026-008914' with a status of 'Active - Routed'. It includes a progress bar with stages: Received, Routed, Under review, Actionable, and Closed. The 'AI READOUT' section contains a summary of the case, mentioning a grey sedan suspected in a car theft. Below this is the 'NARRATIVE' section, which contains a report from the reporter's account. The right sidebar, titled 'Active - Routed', contains workflow actions such as 'Set sub-status', 'Set priority', 'Route this tip', 'Refer to another program', 'Log reward', and 'Close'. The 'Close tip' section includes a dropdown for 'closure outcome' and a 'Reason for closing (required)' field. The 'Close as noise' and 'Pause reporter dialog' options are also visible.

*Closing a case from the workflow actions.*

## Closing as noise

Closing as noise is a separate, narrower action. It is for intake that was not worth routing at all: out of scope, a duplicate or test submission, insufficient information, or a malicious submission. It skips the outcome picker and always lands on Closed, sub-status Noise.

Closing as noise requires program-admin authority. Program staff use Close with an explicit disposition instead.

## Reopening a closed case

Reopening puts a case back to Active, sub-status Pending assignment. It requires program-admin authority and a written reason.

A program admin can reopen only within a reversal window after closure. The default is 72 hours; your program admin can adjust it. Past the window, only a system admin can reopen, and that override is recorded separately from an ordinary reopen. A case revived months later stays visibly different from a routine correction.

## Sub-statuses are configurable; primary statuses are not

The five primary statuses are fixed platform taxonomy. No program can add, rename, or remove one. Sub-statuses are different. Your program admin can add, rename, or retire the sub-statuses under each primary status, to match how your program actually talks about outcomes.

## Related

- [Assigning tips](#)
- [Routing, referring, and cross-program transfers](#)
- [Sensitivity classes and compartments](#)

If a status does not match what you expect, or you need a reversal outside the window above, your program admin can help. For how every status change shows up in the accountability trail, see [Operator accountability](#).

# Rewards: authorize, dispatch, fund

A reward moves through three things you'll deal with: you authorize it against a tip, it gets dispatched to actually pay out, and a reward fund keeps a running balance and history behind both.

## Authorize a reward

There are two ways in:

1. **From the tip itself.** Open its action panel and choose Log reward. Here you can also attach an outcome — arrest, charges filed, property recovered, drugs seized, or another reason — and a case reference, both optional.
2. **From the Rewards page.** Choose New reward and search for the tip by its investigator reference or external case number. This form takes only an amount and a reason; it doesn't capture an outcome or a case reference.

Either way, an amount and a reason are required, and both are program-admin actions. The reason is an internal anti-fraud record, never shown to the tipster. Every reward you create starts in the same place — authorized — whatever else you fill in; there's no way to create one that's already paid.

issued

+ New reward

| REFERENCE       | TIP              | AMOUNT  | STATUS | REASON                                                         | CREATED   |
|-----------------|------------------|---------|--------|----------------------------------------------------------------|-----------|
| RWD-2026-000167 | DEMO-2026-008935 | \$1,000 | issued | Identification led directly to an arrest; reporter had no p... | 9/1/2026  |
| RWD-2026-000165 | -                | \$1,000 | issued | Identification led directly to an arrest; reporter had no p... | 8/31/2026 |
| RWD-2026-000163 | -                | \$1,000 | issued | Identification led directly to an arrest; reporter had no p... | 8/31/2026 |
| RWD-2026-000135 | -                | \$1,000 | issued | Identification led directly to an arrest; reporter had no p... | 7/17/2026 |

\$ Reward Fund

+ Add Funds

Available Balance

\$21000.00

Pending Disbursement

\$4000.00

Recent Transactions

Reward RWD-2026-000167 dispatched

DEMO-2026-008935

+1000.00

Reward RWD-2026-000165 dispatched

+1000.00

Reward RWD-2026-000163 dispatched

+1000.00

Reward RWD-2026-000135 dispatched

+1000.00

Demo reward fund opening balance (seeded)

+\$25000.00

Page 1

Previous

Next

Reward List Report

Independent RWD- ledger — entries survive tip deletion

+ Export CSV

All statuses

All categories

mm/dd/yyyy

to

mm/dd/yyyy

Min \$

Max \$

*Rewards: issued rewards and the fund.*

## What dispatching a reward does

An authorized reward is a commitment, not a payout. Dispatching one is the step that turns it into money moving, and it happens as a single unit: the reward is marked issued and stamped with who dispatched it and when, the reward fund is debited for the exact amount, and — unless the tip has since been deleted — the tipster gets a message in their case thread that a reward is ready to receive.

Dispatch never runs twice on the same reward; a second attempt on one already issued, cancelled, or otherwise moved on is refused outright, so nothing can be paid out twice. It also never runs if the fund can't cover it — a reward larger than the current balance is refused rather than pushed through and left to overdraw the fund.

If the tipster already chose how to receive it, cash or a prepaid card, dispatch uses that choice. Card details are encrypted into the reward record at that point and never appear in an audit row or a report.

From issued, a reward reaches its own end: redeemed once the tipster receives it, or expired if they never do. A reward your program decides not to pay is cancelled instead,

without ever being dispatched. Authorizing, dispatching, and cancelling are each logged — who acted and when — and none of it needs the tipster’s name, because a reward record has nowhere to put one.

## Track the reward fund

The Reward Fund panel on the Rewards page shows what’s actually available: the current balance, and separately, what’s still owed on rewards that are issued but not yet paid out. That second figure comes from the ledger itself, not from a guess based on status alone.

Adding funds is a program-admin action: an amount and an optional note, appended to the same append-only ledger that dispatch debits from. Because both directions run through one ledger, the balance after any single deposit or payout is always traceable to one entry in the transaction list under it, newest first.

## A reward outlives its tip

Deleting a tip doesn’t delete its rewards. The record survives, no longer linked to a tip, and the Rewards list marks it “tip deleted” rather than dropping the row — so a payout stays there to review even after the case behind it is gone.

## Reporting

The reward report at the bottom of the Rewards page filters by status, category, date range, and amount, and exports the result to CSV. Card numbers and PINs are never in it — not filtered out, just never selected in the first place. Exporting needs at least one filter set, and is capped in size by your role and rate-limited per hour.

## Related

- [Getting started as an operator](#)

- [Messaging a tipster](#)
- [Insights, reports, and data export](#)
- [Rewards and demo honesty](#)

If a reward needs to move somewhere this page doesn't take it, or its status doesn't match what you expect, your program admin can look into it.

# Crime of the Week authoring

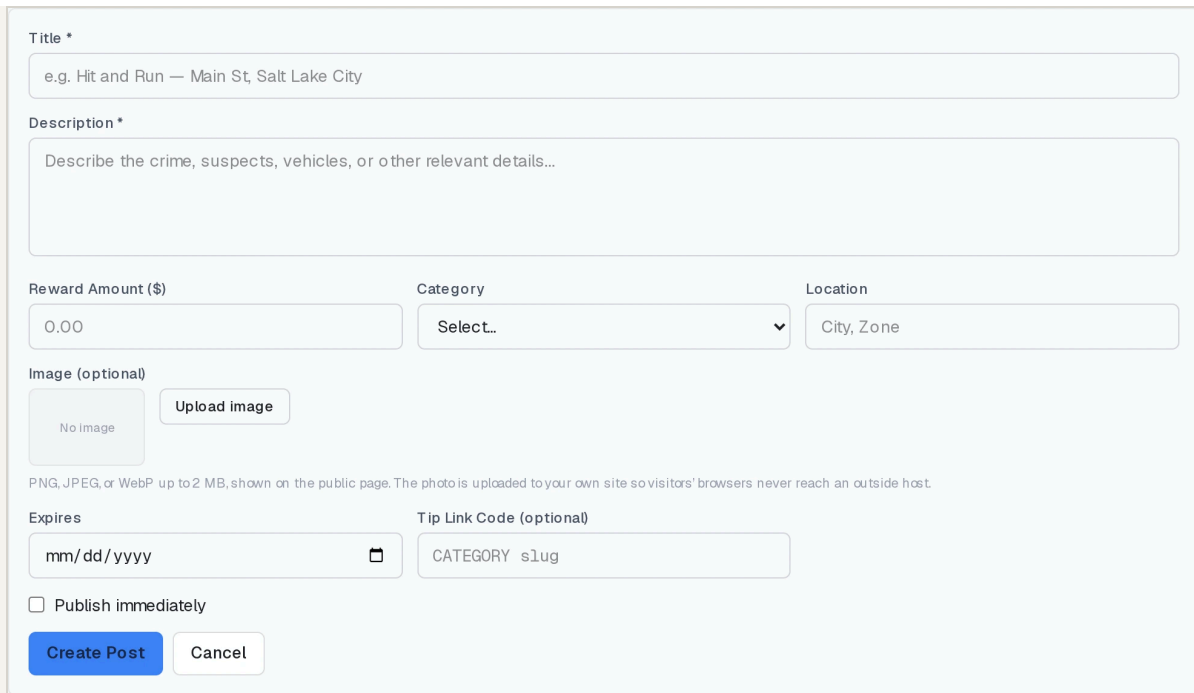
Crime of the Week is a single featured case your program posts to its own public page — a title, a write-up, and optionally a photo and a reward line — to prompt tips on that case. You build and publish it from Settings.

## Where to find it

Open **Settings** and choose the **Crime of the Week** tab. This screen needs administrator access; if you don't see it, ask your program admin.

## Creating a post

1. Select **New Post**.
2. Write a title and a description. Both are required; everything else on the form is optional.
3. Set a reward amount if one is being offered. Any amount above zero adds the reward banner described below.
4. Add a category and a location if they help a reader place the case.
5. Add a Tip Link Code if you want the post's **Submit a tip** button to land on a specific category in your submit form, regardless of the category label shown on the post itself. Leave it blank and the category label is used for that instead.
6. Add a photo if you have one (see below).
7. Set an expiration date if the post should stop showing itself on its own.
8. Check **Publish immediately** to make the post live right away, or leave it unchecked to save a draft you can publish later.
9. Select **Create Post**.



The form is titled "The Crime of the Week" and contains the following fields and controls:

- Title \***: A text input field with the placeholder "e.g. Hit and Run — Main St, Salt Lake City".
- Description \***: A large text area with the placeholder "Describe the crime, suspects, vehicles, or other relevant details...".
- Reward Amount (\$)**: A text input field with the value "0.00".
- Category**: A dropdown menu with the text "Select..." and a downward arrow.
- Location**: A text input field with the placeholder "City, Zone".
- Image (optional)**: A section containing a "No image" button and an "Upload image" button.
- Expires**: A text input field with the placeholder "mm/dd/yyyy" and a calendar icon.
- Tip Link Code (optional)**: A text input field with the placeholder "CATEGORY slug".
- ☐ Publish immediately
- Create Post**: A blue button.
- Cancel**: A button.

Below the image upload section, there is a small note: "PNG, JPEG, or WebP up to 2 MB, shown on the public page. The photo is uploaded to your own site so visitors' browsers never reach an outside host."

*The Crime of the Week form.*

## Adding a photo

Photos are uploaded, not linked. There's no field for pasting an image address: you pick a file from your device, and it's copied onto your own site's storage before anything reaches the public page.

That's deliberate: a pasted link to an outside image host would let that host see every visitor who loads the page, before they've done anything but look. Uploading keeps a visitor's browsing private, because the photo never leaves your own site.

PNG, JPEG, and WebP are accepted, up to 2 MB. Upload a new file to replace an existing photo, or remove it to go back to no photo at all.

## The reward banner

Set a reward amount above zero and the public page shows a banner naming that amount, with a line reminding the reader that reporting stays anonymous. Leave the

amount blank, or at zero, and the banner doesn't appear. The rest of the post looks the same either way.

## Publishing

Only the most recently published post that hasn't expired shows on the public page at any one time. Publishing a new post doesn't take an older one down automatically: if you leave a previous post marked live, it keeps that label in your list, but visitors only ever see whichever live post was published most recently.

### Note

If you want an old post fully retired rather than just superseded, unpublish it yourself. Otherwise it sits in your list marked live, even though the public page has already moved on to whatever you published after it.

## Managing existing posts

From the post list, the eye icon publishes or unpublishes a post, the link icon copies the public page's address, and the external-link icon opens that page for you to check. Deleting a post removes it permanently; it doesn't touch any tip or case record, since a Crime of the Week post is a marketing surface, separate from whatever tips it prompts.

## Related

- [Getting started as an operator](#)
- [Signing in and two-factor authentication](#)

The same reasoning behind requiring uploads instead of pasted links — keeping a visitor's browsing from reaching an outside host — is described for reporters in [Anonymity](#).

# Insights, reports, and data export

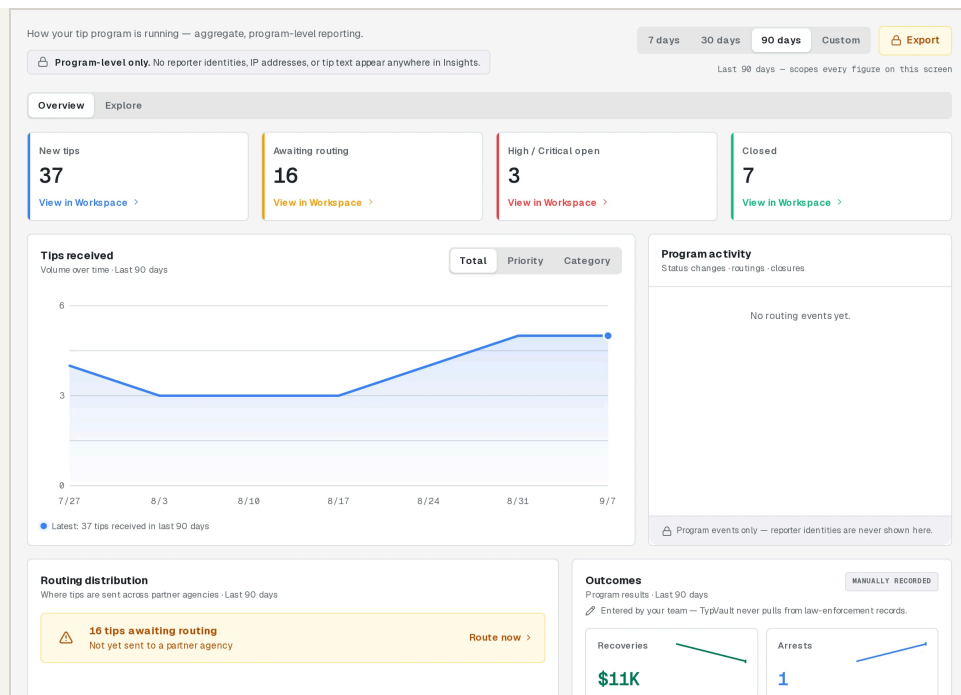
Insights is where you see how your program is doing overall — case volume, how long cases sit at each stage, and program outcomes — and where you export data out of TypVault. The old Reports screen is gone; everything it showed now lives here, on the same page.

## What Insights shows

- **Volume**, broken down by category, priority, status, disposition (how a case was closed), receiving agency, org unit, or how the tip came in.
- **Lifecycle**, on the Explore tab: how long cases sit at each stage, and where cases drop off before reaching a close.
- **Outcomes**: dollars recovered, arrests, cases cleared, and rewards issued.

Every number on this page is a total or a breakdown across many cases. Insights never drills down to one reporter's identity, an IP address, or a case's narrative. Those live on the case itself, not here.

Insights is a program admin view. If it's missing from your navigation, that's your role, not a fault.



*Insights for the last 90 days.*

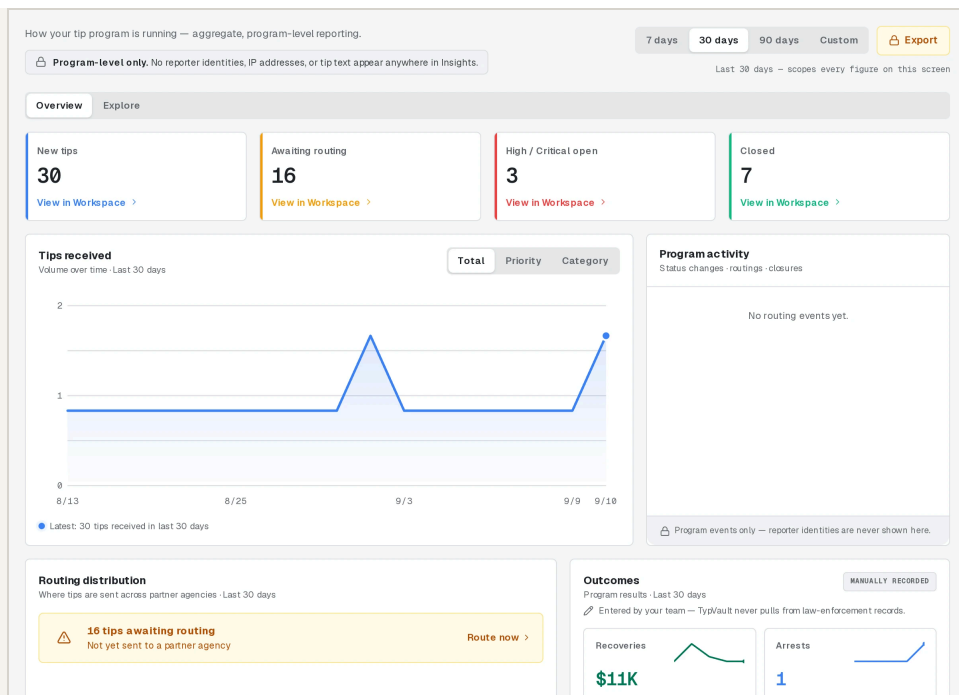
## Outcomes are entered by hand

Recoveries, arrests, and cleared cases don't show up here on their own. An operator records them on the case itself, whenever the numbers are known. Nothing arrives automatically from an outside case-management system. If a number is missing, no one has entered it yet.

*The Outcomes tab of a tip.*

Anyone viewing Insights as a program admin can pull the on-screen breakdown out as a CSV.

- This file holds totals only: a category, a label, and a number. It can't carry a reporter's identity or a case's narrative, because it never reads either.



*Reports.*

## Exporting tip records for legal discovery

Pulling one row per case for a legal-discovery production works differently: your reason alone isn't enough to run it. A different supervisor has to approve the request first. The same person can't request and approve their own export.

1. In the same Export dialog, choose **Tip records (discovery)**.
2. Set a scope: a start date, and optionally a status, category, or priority.
3. Write a reason and submit. The request sits pending.
4. Once a different supervisor approves it, run it from the requests list to download the file.

Even here, each row holds case-level fields only: a case reference, category, status, priority, dates, and reward status. The narrative and the reporter's identity never leave through either export path.

### **Note**

An approval is single-use. Once you run the export, that approval is spent. A second production needs its own request and its own approval.

## Every export is on the record

Whichever path you use, TypVault writes the export to the audit trail — your name, your reason, and what was pulled — before the data is read, so the record exists even if the download never finishes.

## Related

- [Access Review for supervisors](#)
- [Statuses, sub-statuses, and the case lifecycle](#)
- [Sensitivity classes and compartments](#)
- [Operator accountability](#)

If a number here looks wrong, or you need an export this page doesn't cover, ask your program admin.

# Agencies, MOUs, and outreach

The agency directory is where you track law-enforcement agencies, their MOU status, and every contact your program has had with them: one record per agency, one shared history the whole team can see.

## The agency directory

Each agency record carries its name, county, agency type, phone numbers, website, and one or more contacts; mark one contact primary so it's the first name anyone sees. Search the directory by agency name, contact name, or county, and narrow it by MOU stage. A program admin can add a new agency by hand, or bring in a batch from a spreadsheet file; the same list can be pulled back out as a spreadsheet for a backup or a handoff.

Taking an agency out of the active list doesn't erase it. The record is set aside rather than deleted, so its contacts, MOU documents, and history stay in place if it's ever needed again.

| Search by name, contact, or county...                            |               |                        |                 | ▼                                | All statuses | ▼ | 📄 Export | 📄 Import |
|------------------------------------------------------------------|---------------|------------------------|-----------------|----------------------------------|--------------|---|----------|----------|
| AGENCY                                                           | COUNTY        | STATUS                 | PRIMARY CONTACT |                                  |              |   |          |          |
| <b>Capital City University Police</b><br>CAMPUS                  | Metro County  | discussion in progress | —               | tips@capitalcity-univ-pd.example | >            |   |          |          |
| <b>Eastvale Police Department</b><br>POLICE                      | Metro County  | initial outreach sent  | —               | tips@eastvale-pd.example         | >            |   |          |          |
| <b>Fairhaven Police Department</b><br>POLICE                     | Harbor County | declined               | —               | tips@fairhaven-pd.example        | >            |   |          |          |
| <b>Federal Field Office — Metro Division</b><br>FEDERAL          | Metro County  | draft sent             | —               | tips@federal-fo.example          | >            |   |          |          |
| <b>Fenwick Police Department</b><br>POLICE                       | Harbor County | not started            | —               | tips@fenwick-pd.example          | >            |   |          |          |
| <b>Harbor County DA Investigations</b><br>OTHER                  | Harbor County | signed by program      | —               | investigations@harbor-da.example | >            |   |          |          |
| <b>Harbor County Sheriff's Office</b><br>SHERIFF                 | Harbor County | active                 | —               | tips@harbor-so.example           | >            |   |          |          |
| <b>Harbor Transit Police</b><br>OTHER                            | Harbor County | initial outreach sent  | —               | tips@harbor-transit-pd.example   | >            |   |          |          |
| <b>Lakemont Police Department</b><br>POLICE                      | Harbor County | not started            | —               | tips@lakemont-pd.example         | >            |   |          |          |
| <b>Lakemont Tribal Police</b><br>TRIBAL                          | Harbor County | not started            | —               | tips@lakemont-tribal-pd.example  | >            |   |          |          |
| <b>Metro County Cold Case Task Force</b><br>OTHER                | Metro County  | negotiating            | —               | coldcase@metroville-pd.example   | >            |   |          |          |
| <b>Metro County Fire Marshal — Arson Investigations</b><br>OTHER | Metro County  | not started            | —               | arson@metro-firemarshal.example  | >            |   |          |          |
| <b>Metroville Police Department</b>                              |               |                        | —               |                                  | >            |   |          |          |

*The agency directory.*

# Tracking an MOU

Every agency carries an MOU stage: not started, initial outreach sent, discussion in progress, draft sent, negotiating, signed, fully executed, active, declined, or withdrawn. Once one is signed, the record also holds an effective date and an expiration date. An agency whose MOU is fully executed and due to expire within 60 days gets a visible flag on the directory, so a renewal doesn't sneak past.

## Generating and uploading a document

Two separate actions live on an agency's page, and they don't do the same thing. Generate MOU builds a Word document from a standard template, filled in with that agency's details and your program's name; add the executive director's name for the signature line, or leave it blank. This only produces a file to review and send; it doesn't change the agency's MOU stage by itself.

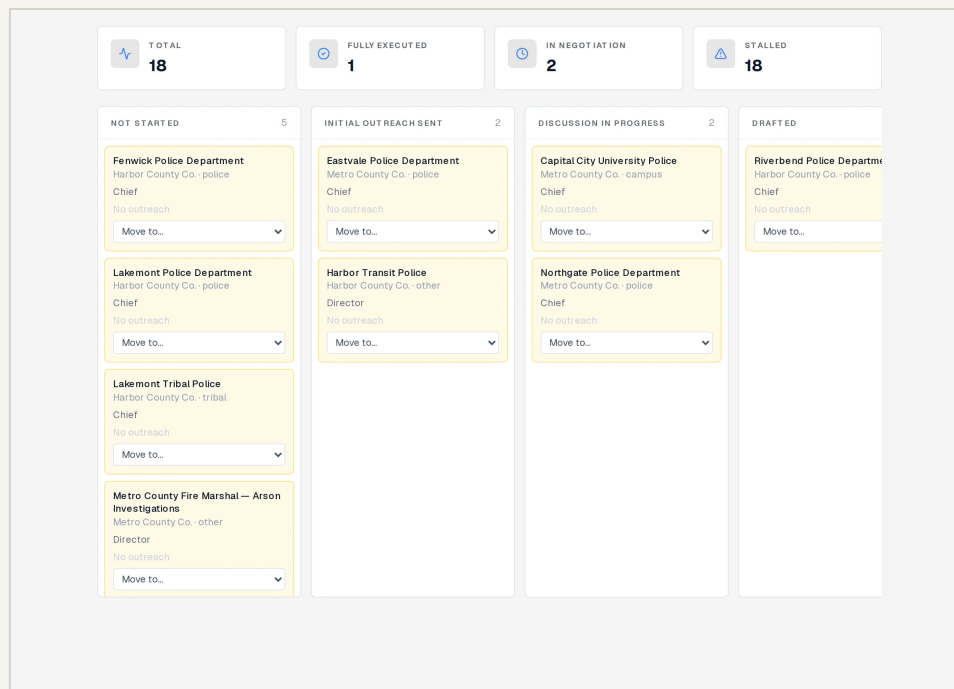
Once a copy comes back signed, upload it as a PDF or Word file, up to 10 MB. Only those two file types are accepted, and the file's actual contents have to match what it claims to be: renaming a different kind of file to look like a PDF doesn't get past the check. If your program has virus screening enabled, uploads are scanned before they are kept and a file that does not pass is refused, and nothing about it is written to the agency's record.

Each accepted upload becomes its own numbered version, and you can set the MOU stage at the same time you upload it. Signed MOUs are kept in storage that isn't publicly reachable; opening one later goes through a link generated for that one request, and it always comes back as a file to save, never opened directly in the browser.

## The MOU pipeline board

The pipeline board lays every agency out by MOU stage, side by side, with totals for the whole program: how many agencies overall, how many are fully executed, how many are in active negotiation, and how many have gone 30 days or more without contact. It's a view for seeing where things stand at a glance; moving an agency to a new stage is a

separate, deliberate action: choose a destination on that agency's card and confirm the move. Adding a short reason when you do carries it straight into that agency's contact history, so the move itself is on the record.



*The MOU pipeline board.*

## Tracking contact with an agency

Every call, email, meeting, letter, or visit involving an agency gets a short entry on its record: the channel used, whether it went out from your program or came in from the agency, a plain summary, and, if there's a next step, a follow-up date. An entry with a follow-up date that's arrived shows up flagged on that agency's page until it's handled.

That per-agency history also feeds a program-wide view: every entry from every agency in one list, which you can narrow down by agency, by who logged it, by channel, or by date range. It's the place to look when the question is about a pattern across the whole directory, not just one agency's file.

## Who can do this

Any program staff member can browse the directory, open an agency's page, and view the pipeline board. Adding or editing an agency, adding or editing a contact, generating or uploading an MOU, moving a pipeline stage, and adding a contact entry all need program admin standing.

## Related

- [Getting started as an operator](#)
- [Routing, referring, and federation](#)
- [Statuses and case lifecycle](#)

Questions about a specific agency's MOU or contact record go to your program admin.

# Recipient portal administration

A recipient's portal login lets an agency you route tips to sign in and see its own cases, without waiting on your team for every update. You invite that login, and you can end it, from **Recipient Access** in the operator console.

## Inviting a recipient

**Recipient Access** lists every active recipient in your directory. Select one to open its panel: every login it already has, and a form to add another.

1. Open **Recipient Access**.
2. Select the recipient from the list.
3. Enter the contact's email, and a name if you have one.
4. Choose **Send invite**.

TypVault doesn't send the link on its own. Copy the one-time link it hands you and deliver it yourself, the way your program already reaches that agency.

The link works once, and only for 72 hours. Opening it lets the agency set its own password; the login activates the moment that happens, not before.

Until it's used, the login sits as **invited**. Once redeemed, it turns **active**. If you end it later, it turns **revoked**, a state it stays in.

### Note

An email address can belong to only one recipient's portal at a time. If you try to invite the same address into a second recipient's portal, TypVault blocks it. Invite it under a different address instead.

Invite a routed agency to its own login, where it sees only the tips routed to it. An invite link activates the account once redeemed; revoke ends access immediately. ⓘ

|                                                                   |                                              |                                            |
|-------------------------------------------------------------------|----------------------------------------------|--------------------------------------------|
| > Cold Case Unit (Demonstration)                                  | ccu                                          | cold case unit                             |
| > Federal Field Office (Demonstration)                            | ffo                                          | federal                                    |
| > Harbor County Sheriff's Office                                  | hcs0                                         | sheriff                                    |
| ▼ Metroville Police Department                                    | mfd                                          | local pd                                   |
| tips@metroville-pd.example · Metroville PD Portal active ⓘ Revoke |                                              |                                            |
| <input type="text" value="agency email"/>                         | <input type="text" value="name (optional)"/> | <input type="button" value="Send invite"/> |
| > State Patrol (Demonstration)                                    | stp                                          | state patrol                               |

*Recipient Access: inviting an agency.*

## What the login can see

Every case routed to a recipient shows its portal exactly four fields: the reference your team assigned, the category, the status, and when it came in.

The login never sees the narrative, location detail, attached files, or your team's working notes. That boundary is fixed, not a setting: there's no configuration that widens it.

Case sensitivity narrows it further. A case walled into Sealed, Internal Affairs, or Intelligence never reaches a portal, even if it's routed there. Only cases left in the default Unrestricted class are eligible to reach one at all.

For what this looks like from the agency's side, see [Receiving tips at your agency](#).

## Revoking access

Every active login shows a **Revoke** control next to it. Choosing it turns that login off immediately: the agency can't sign in again, and the change takes effect right away.

A still-pending invite — one the agency hasn't opened yet — doesn't show a **Revoke** control. Left alone, it expires at the 72-hour mark.

Revoke can't be undone from this screen, and the address doesn't get a clean slate afterward. Invite it again later, and TypVault blocks the attempt: that address stays tied to the revoked login, not to a new one.

Every invite and every revoke is logged to your program's audit trail, with who did it and when. That entry can't be edited or removed afterward, by anyone.

## Related

- [Routing, referring, and cross-program transfers](#)
- [Sensitivity classes and compartments](#)
- [Access Review for supervisors](#)
- [Receiving tips at your agency](#)

Adding a new agency to your directory in the first place isn't done here. That happens under **Agencies** in the operator console, well before there's a recipient here to invite.

# Sensitivity classes and compartments

Every case carries a sensitivity class. Most stay in the default class, Unrestricted, and follow your program’s ordinary routing and role rules. A program admin can move a case into one of three restricted classes instead: Sealed, Internal Affairs, or Intelligence. Each walls the case off to a small group of staff, regardless of anyone’s rank.

## What the restricted classes are for

- **Sealed:** a case held back from the general queue for any high-sensitivity reason.
- **Internal Affairs:** a case that involves program or agency personnel, kept separate from the people it concerns.
- **Intelligence:** a case restricted to staff working intelligence matters.

All three work the same way: opening one requires either membership in that specific compartment, or being the operator the case is currently assigned to. Anyone else is shut out, whether they hold the highest role in your program or the newest account.

⚠ Need-to-know membership. A walled case (sealed, internal affairs, intelligence) is reachable only by operators in its compartment — rank-independent: even a Sys Admin without membership cannot open it. Changes apply immediately and are written to the audit trail.

| OPERATOR                    | Sealed                   | Internal Affairs         | Intelligence             |
|-----------------------------|--------------------------|--------------------------|--------------------------|
| MFA Walkthrough<br>Operator | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Alyn Grey<br>Sys Admin      | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Dev Admin<br>Sys Admin      | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

*Compartments.*

# Membership, not rank

Compartment membership is a separate list from your role, and a program admin manages it person by person. Adding someone to Sealed doesn't add them to Internal Affairs, and neither follows from being a program admin or sys admin.

A sys admin outranks every other operator role, but rank buys nothing here: without active membership in a compartment, a sys admin can't open a case walled into it, and neither can anyone else outside that group.

## Reclassifying a case

Moving a case between classes is a program admin task, done from the sensitivity control on the case header in your workspace.

1. Open the case in your workspace.
2. Find the sensitivity control on the case header. It shows the current class, with a lock icon on the three restricted ones.
3. Choose a new class. Moving into Sealed, Internal Affairs, or Intelligence asks you to confirm, since it walls the case away from most of the program's other staff going forward.
4. Confirm. You can only reclassify into a compartment you already belong to: you can't wall a case into a group you're not a member of yourself. Moving a case back to Unrestricted doesn't require membership in anything.

### Caution

Reclassifying takes effect immediately for everything already tied to the case: analysis notes, related-case matches, the case's own history. There's no separate step to finish walling it off, and no gap where the old, wider access still works.

## What a blocked operator sees

If a case is walled off and you're neither a compartment member nor its assigned operator, it won't appear in your queue. Opening it directly — a saved link, a search hit, a reference from another case — fails the same way a nonexistent case would. TypVault never distinguishes “this case exists but you can't see it” from “this case doesn't exist,” because that distinction would itself tell you something you're not supposed to know.

If you believe a case should be visible to you, ask your program admin to add you to its compartment, or to check whether it's assigned to you.

## A history of every change

Every compartment membership grant or removal, and every case reclassification, is logged with who made it and when, so a program admin can trace how a case's access has changed over time.

## Related

- [Getting started as an operator](#)
- [Reading a tip](#)
- [Routing, referring, and cross-program transfers](#)
- [How TypVault protects you](#)

If a class or compartment does not match what you expect, ask your program admin.

# Access Review for supervisors

Access Review is the cross-operator view of who accessed tip data, when, and why: one row per operator, with a drill-down into the exact events behind it.

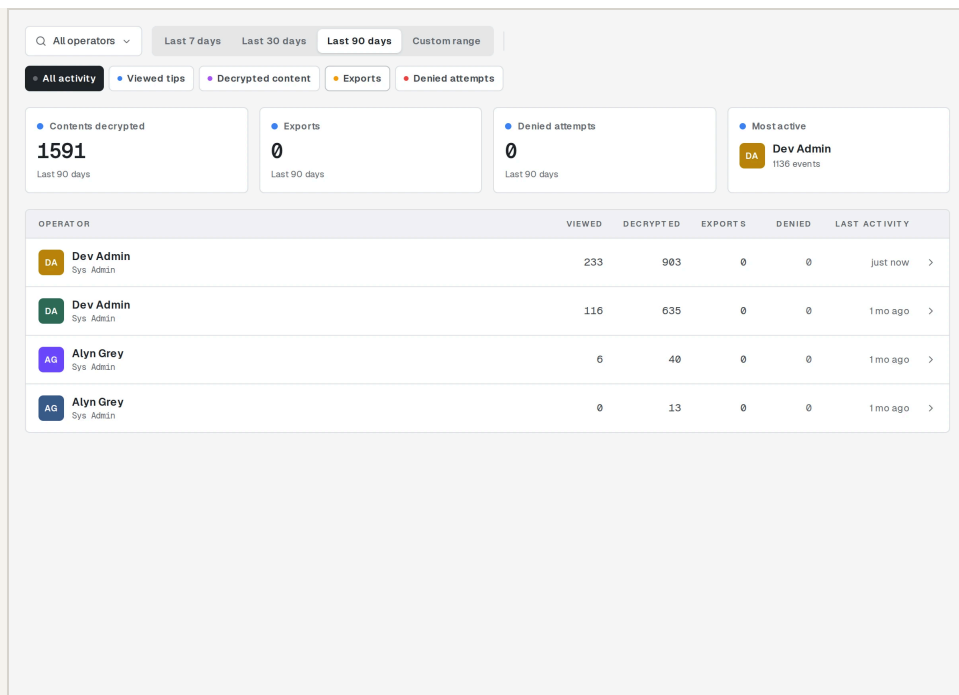
Where a single tip's audit view (see [Reading a tip: the audited decrypt](#)) shows everyone who has touched that one case, Access Review flips the angle: pick a time window and see every operator's activity across every tip, side by side.

## What the summary shows

For each operator in the window, the table shows four counts:

- **Viewed:** tips opened, summaries expanded, the inbox opened.
- **Decrypted:** sealed content actually opened for reading.
- **Exports:** data pulled out for reporting or legal disclosure.
- **Denied:** access attempts the system blocked.

Alongside the table, a summary strip totals decrypts, exports, and denials for the whole program, and names whoever was most active in the window.



*Access Review, filtered to the last 90 days.*

## Running a review

1. Open **Access Review** from the admin section.
2. Set a date range: the last 7, 30, or 90 days, or a custom span.
3. Optionally narrow to one operator by name.
4. Optionally filter to one activity type: viewed, decrypted, exports, or denied.

The table re-sorts by total activity, busiest operator first.

## Drilling into one operator

Select any row to expand it into that operator's individual events for the window: what happened, roughly when, and — where one was recorded — a stated reason such as opening a tip, printing it, or looking up its location.

Long lists load in pages rather than all at once; a **Load more** control appears when there's more to see.

## The unusual-volume flag

An operator whose total is well above the rest of the team's for that window gets a flag next to their name. It's a statistical comparison, not a conclusion: a busy week, a reassignment, or a single large case can all produce one honestly. Use the event log underneath to check the individual reasons before treating it as anything more than a prompt to look closer.

### Note

The threshold moves with your program's own activity. A small team's baseline looks nothing like a large one's, so the flag compares each operator only against your team's own numbers, not a fixed number.

## What never appears here

Reporter activity is never logged, so it never appears in Access Review. This surface records what operators and the system did, not what a tipster wrote or did on their end.

## Why the trail can't be cleaned up afterward

Every access row is written before the record it describes is even shown: a failed or denied attempt still leaves one. Once written, no operator role can edit or remove a row. That includes running an access review itself: opening this screen is its own logged event, so a supervisor's own reviewing is part of the same accountable trail.

## Who can run a review

Access Review needs the audit-review permission, held by program admins and system admins by default. Anyone else who opens the page sees a plain notice instead of the screen: never a broken page, never partial data. If you need this view and don't have it, ask your program admin.

## Related

- [Reading a tip: the audited decrypt](#)
- [Getting started as an operator](#)
- [Signing in and two-factor authentication](#)

The same accountability guarantee is described for reporters, in their own words, on [Operator accountability](#).

# Program administration: users, roles, org units, retention, taxonomy

Program administration covers the platform-level settings that keep a program running: who has an account, what each role can do, how your agency hierarchy is organized, how long closed tips are kept, and how your status labels work. These are administrative controls, not day-to-day tip handling: tip actions live in your workspace.

## Adding and managing users

A system admin adds a new operator account from the user list: an email address, a display name, and a starting role. There's no separate invitation step: the account is created directly and becomes usable the next time that person signs in.

From the same list, a system admin can:

1. Change a person's role.
2. Suspend or deactivate an account, or reactivate one that was deactivated.
3. Open a person's detail page to review their sign-in history and language proficiencies.

Only a system admin can create a user, change a role, or change status. A program admin's authority stops short of these actions.

A system admin can't change their own role or deactivate their own account through this screen; that's blocked outright, so any role change always has a second person behind it. Every user created, edited, or deactivated is written to the audit trail.

### Note

Deactivating an account doesn't erase it. History and prior actions stay associated with it. The account is no longer usable to sign in.

3 users ↻ [+ Add user](#)

| Name / Email                                           | Role          | Status | Last login | Actions                             |
|--------------------------------------------------------|---------------|--------|------------|-------------------------------------|
| <b>Alyn Grey</b><br>a@typvault.org                     | System admin  | Active | Never      | <a href="#">✎</a> <a href="#">✖</a> |
| <b>Dev Admin</b><br>admin@typvault.org                 | System admin  | Active | Never      | <a href="#">✎</a> <a href="#">✖</a> |
| <b>MFA Walkthrough</b><br>mfa-walkthrough@typvault.org | Program staff | Active | Never      | <a href="#">✎</a> <a href="#">✖</a> |

*Users.*

## The two role tiers

TypVault's roles come in two independent tiers. The first is your operator-wide role: program staff, program admin, or system admin, each a step up in authority from the last, with system admin as the full-access tier reserved for platform administration. The second tier is separate and applies only to staff at a recipient agency your program routes tips to: an agency user acts on tips sent to their agency, and an agency admin also manages that agency's own members. The two tiers don't overlap: a recipient-agency role carries no operator-wide authority, and vice versa.

A program admin can also fine-tune what each non-locked role is allowed to do, action by action — routing, reassigning, closing a tip, requesting an export, and similar — from the roles and permissions screen. The system admin tier itself can't be narrowed; it's the platform's break-glass authority and always keeps full access.

🔒 What each role may do. Changes apply immediately and are written to the audit trail. The action catalog is fixed in code; the grants are yours to configure. Sys Admin is the break-glass role — always full, not configurable.

| ACTION                                                                                                                                              | Sys Admin<br>OPERATOR | Program<br>Admin<br>OPERATOR | Operator<br>OPERATOR | Agency<br>Admin<br>RECIPIENT | Agency<br>User<br>RECIPIENT |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|------------------------------|----------------------|------------------------------|-----------------------------|
| CASE VISIBILITY                                                                                                                                     |                       |                              |                      |                              |                             |
| <b>Read all cases</b><br>See every case program-wide — not only those assigned or reachable by membership.                                          | 🔒                     | ✓                            | —                    | —                            | —                           |
| <b>Act on all cases</b><br>Take workflow actions on any case program-wide, not only assigned ones.                                                  | 🔒                     | ✓                            | —                    | —                            | —                           |
| CASE WORKFLOW                                                                                                                                       |                       |                              |                      |                              |                             |
| <b>Set sub-status</b><br>Advance a case within its current stage (eg mark under review).                                                            | 🔒                     | ✓                            | ✓                    | ✓                            | —                           |
| <b>Set priority</b><br>Change a case's priority.                                                                                                    | 🔒                     | ✓                            | ✓                    | ✓                            | —                           |
| <b>Route to recipient</b><br>Route a case to a recipient agency or group.                                                                           | 🔒                     | ✓                            | ✓                    | —                            | —                           |
| <b>Refer to peer program</b><br>Federate a case to a paired program on another deployment.                                                          | 🔒                     | ✓                            | —                    | —                            | —                           |
| <b>Reassign</b><br>Change which operator a case is assigned to                                                                                      | 🔒                     | ✓                            | ✓                    | —                            | —                           |
| <b>Close case</b><br>Move a case to a resolved/unresolved closure status.                                                                           | 🔒                     | ✓                            | ✓                    | —                            | —                           |
| <b>Close as noise</b><br>Dispose a non-actionable submission.                                                                                       | 🔒                     | ✓                            | —                    | —                            | —                           |
| <b>Reopen case</b><br>Return a closed case to active.                                                                                               | 🔒                     | ✓                            | —                    | —                            | —                           |
| <b>Pause reporter dialog</b><br>Stop and restart the two-way conversation on a case, without closing it. Both edges are written to the audit trail. | 🔒                     | ✓                            | ✓                    | —                            | —                           |
| <b>Message reporter</b><br>Send messages to the anonymous reporter on a case — the program's direct two-way channel.                                | 🔒                     | ✓                            | ✓                    | —                            | —                           |

*Roles and permissions.*

## Org units: the agency hierarchy

Org units are the tree your tips route through: precinct under borough under department, or whatever hierarchy fits your program. A program admin creates, renames, and moves units anywhere in the tree. A delegated admin — an agency admin scoped to one branch — can do the same, but only within their own branch; moving a unit outside that branch, or restructuring the branch's own boundary, stays a program-admin action.

Anyone signed in, from either role tier, can see the whole tree — it's routing structure, not tip content — even if they can't edit any of it.

A The routing hierarchy a case can be sent to. Nest units (precinct under borough under bureau); routing to a parent reaches its whole subtree. A sub-org administers only its own subtree.

List Chart

| CODE                       | DISPLAY NAME  | PARENT      |
|----------------------------|---------------|-------------|
| precinct_19                | 19th Precinct | — Top level |
| <a href="#">+ Add unit</a> |               |             |

| HIERARCHY                          |             |                            |
|------------------------------------|-------------|----------------------------|
| All Local Law Enforcement local_le | — Top level | <a href="#">Deactivate</a> |
| Specialized Units specialized      | — Top level | <a href="#">Deactivate</a> |

*Org units.*

## Retention: review before purge

Two settings control how long a closed tip is kept: how many days pass before it's due for review, and how many hours after closing it can be reopened without a system-admin override. A system admin sets both; the retention window can't be set below a 1,000-day floor.

When a tip reaches its retention window, TypVault doesn't remove it outright: it flags the tip for review instead. An authorized operator then decides: reaffirm it, which resets the clock, or clear it for permanent removal. Only a cleared tip is ever actually destroyed, and destroying it also destroys the key that protected its contents, so nothing left in the live system can read what it said.

The audit trail itself isn't part of this window. It's kept on a fixed schedule of its own, because no application role can edit or delete an audit row at all.

The screenshot shows a web application interface for 'Retention' settings. At the top, there is a navigation bar with links: Program Settings, Jurisdictions, Sub-statuses, Recovery Categories, MOU Templates, and Crime of the Week. Below this, the 'Retention' tab is selected, with sub-links for Users and Federation. A warning icon and text state: 'Data-retention windows. The retention worker runs on these (safe defaults apply if unset).' There are three main sections: 1. 'Tip retention' with a text input '1825' and a 'days' unit, a 'Save' button, and a variable 'retention.tip\_retention\_days'. 2. 'Reopen window' with a text input '72' and a 'hours' unit, a 'Save' button, and a variable 'retention.reopen\_window\_hours'. 3. 'Audit log -- fixed' with a description: 'The audit trail is append-only and retained for the CJIS minimum (≥ 12 months). No application role can delete or modify audit rows -- this is enforced structurally (no DELETE/UPDATE grant), not a setting.'

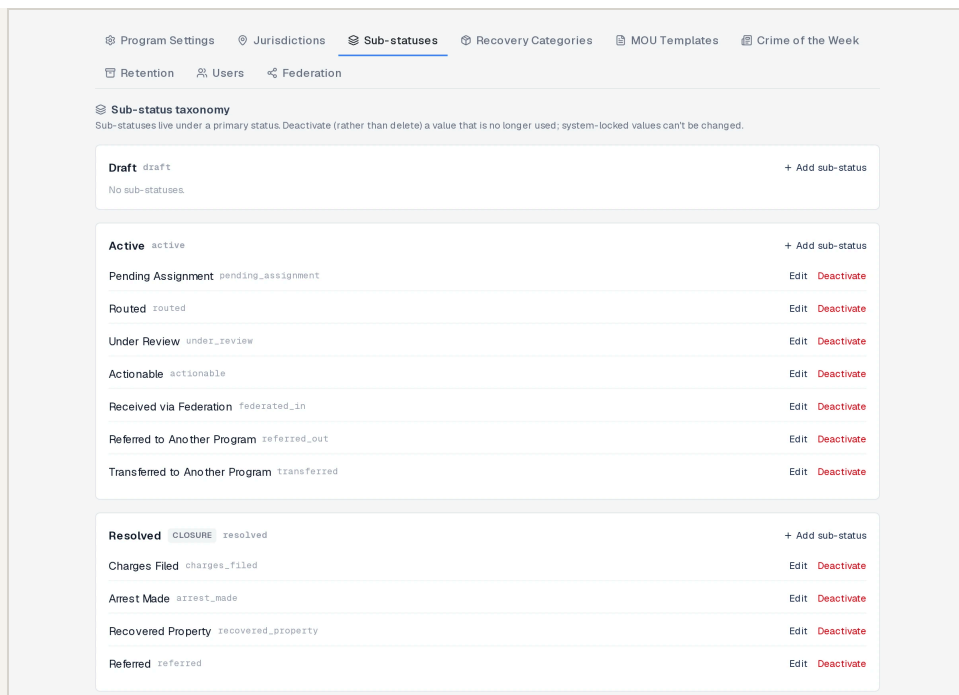
*Retention settings.*

## Status taxonomy: shape it to your program

Every tip's five primary stages — draft, active, resolved, unresolved, closed — are fixed platform taxonomy; no program can add, rename, or remove one. What you can shape is the sub-status underneath each stage: the specific labels your program actually uses, such as “awaiting warrant” or “confirmed duplicate.”

A program admin adds a sub-status under any stage, marks whether reaching it counts as closing the tip, and sets the order it displays in. Retiring one doesn't remove it outright: a sub-status already in use on a tip is deactivated instead, so nothing on an existing tip goes missing. A small number of sub-statuses are locked platform defaults and can't be edited at all.

For how these statuses move a tip through its day-to-day lifecycle, see [Statuses, sub-statuses, and the tip lifecycle](#).



*Sub-statuses, shaped to your program.*

## Related

- [Statuses, sub-statuses, and the tip lifecycle](#)
- [Sensitivity classes and compartments](#)
- [Signing in and two-factor authentication](#)
- [Access Review for supervisors](#)

If something here doesn't match what you see in your program, ask your system admin, or see [How TypVault protects you](#) for the underlying guarantees.