PolySwarm Playbook Templates

Six ready-made playbooks for Sekoia.io

PRODUCT	Playbook templates for the PolySwarm module
MODULE VERSION	1.0.0
PLATFORM	Sekoia.io Defend and Intelligence Center
AUDIENCE	SOC analysts and playbook authors
COMPANION	PolySwarm for Sekoia User Guide
SUPPORT	customersuccess@polyswarm.io
PUBLISHED	2026-11-13
CLASSIFICATION	Public

What this is

Six ready-made playbooks for Sekoia.io, each wired around the PolySwarm integration. A template is a complete workflow somebody has already built: the trigger, the actions, the branches and the text written back onto the alert. You import one, point it at your PolySwarm account, and it runs.

Nothing about an account is baked into a template, so importing one never carries anyone else's key or community.

Before you start

These assume the PolySwarm integration is installed and an account is connected. Both are covered in **PolySwarm for Sekoia User Guide**, which also documents the individual actions these templates are built from, what each one costs, and what leaves your environment. Read that first if you have not connected an account yet.

We've submitted six playbook templates to Sekoia's Playbooks Catalog. Once Sekoia publishes them, you'll find them under `Start from a template`. Until then, ask customersuccess@polyswarm.io for the template files. Six PolySwarm templates cover the jobs below.

Adding any of these, once

The steps are the same for all six, so they are here rather than repeated under each one.

1. Go to `Playbooks` and create a new playbook.
2. On the empty canvas, click `Start from a template`.
3. Search the Playbooks Catalog for `PolySwarm`, or filter by the `Enrichment` tag. Pick the template you want. Its workflow fills the canvas.
4. Click each PolySwarm node and choose your connected account in the configuration panel. This is the only configuration a template needs from you. If no account is listed, go back and connect one first.
5. Check the comment text on each branch. It is written to be useful as it stands, but it is ordinary text and yours to reword.
6. Turn the playbook on.

A template that was imported but never pointed at an account will fail on its first PolySwarm node. That is the commonest mistake, and the error says the account is missing rather than anything about PolySwarm.

One cosmetic thing you will notice in the screenshots below. In the run view, some nodes show a generic gear icon rather than the PolySwarm logo. That is the interface, not a sign the node is wrong or unconfigured, and it survives a reload. Read the node name rather than the icon.

Enrich an alert with PolySwarm

The job. “An alert just landed with a file hash on it and I need to know whether anyone has seen this thing before I spend an hour on it.”

When to use it. On any alert carrying a file hash, including as the first step of a triage playbook that runs on everything. The lookup is read only and spends no scanning quota, so running it on every alert costs nothing.

When not to. There is no real reason not to. If the hash is missing from the alert the playbook simply has nothing to look up.

Search the catalog for: Enrich an alert with PolySwarm.

The template branches three ways and comments a different message on each. The unknown branch is wired separately on purpose: an artifact nobody has seen is not the same as one nobody flagged.

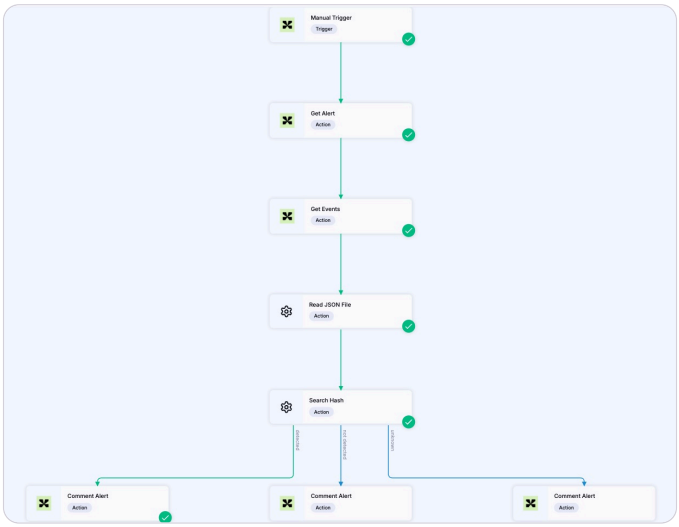

Figure 1: The enrichment template after a run, with the detected branch taken.



Figure 2: The comment it writes back onto the alert, carrying the engine counts and the family.

If something looks wrong. An unknown result on an artifact you are sure is known is almost always the community setting on the account rather than the playbook.

Detonate an unknown URL with PolySwarm

The job. “Nothing knows anything about this URL. I want to see what it actually does, without sitting and waiting for it.”

When to use it. When a lookup comes back unknown and the URL matters enough to spend on. The template checks first and only detonates if nothing is known, so it does not pay for an answer that already exists.

When not to. Not on every alert. Detonation spends against your plan, and unlike a lookup it cannot be undone: what you submit is shared with every engine in your community. Think before pointing this at a customer’s internal URL.

Search the catalog for: Detonate an unknown URL with PolySwarm.

It comments that the run has started rather than waiting for it, because a detonation takes minutes. The result arrives through the next template.

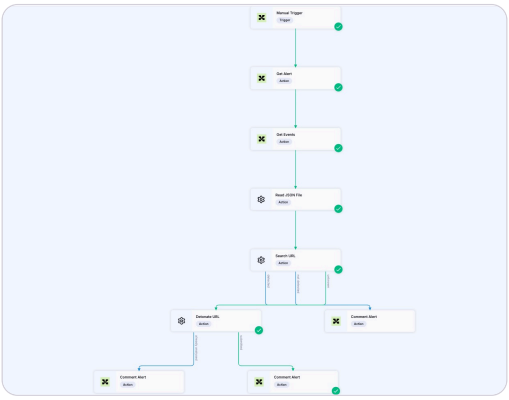

Figure 3: The detonation template. The lookup returned unknown, so the URL went to the sandbox.

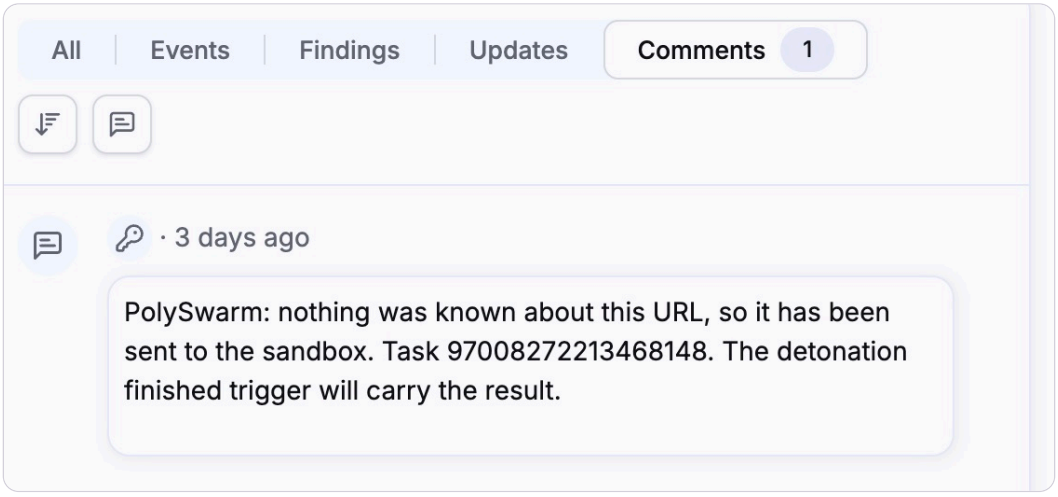

Figure 4: The comment confirming the run has started, with the task to follow it by.

If something looks wrong. If it comments that a report already existed, that is the template working: somebody had already detonated that URL and it reused the answer instead of paying again.

Handle a finished PolySwarm detonation

The job. “Something I sent to the sandbox has finished. I want the report pulled automatically rather than remembering to go back for it.”

When to use it. Alongside the template above, or any time you submit detonations. It starts itself when a detonation finishes, so nobody has to watch a queue.

When not to. It is deliberately unfinished: it fetches the report and branches on the verdict, and leaves the response step to you. Add your own next step before relying on it.

Search the catalog for: Handle a finished PolySwarm detonation.

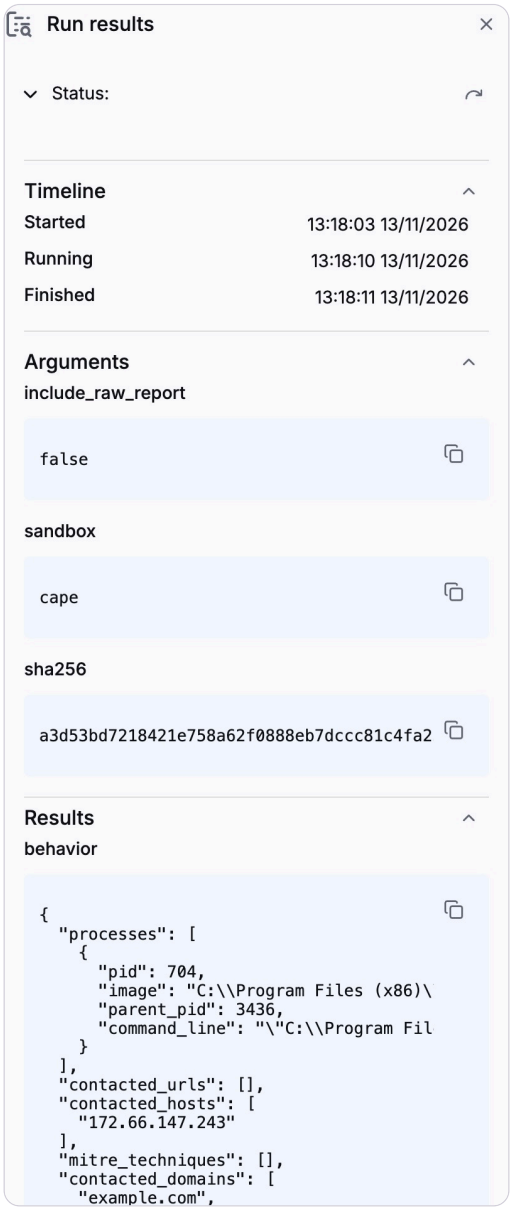

Figure 5: The report it fetched. This run took the not malicious branch, with the behaviour visible underneath.

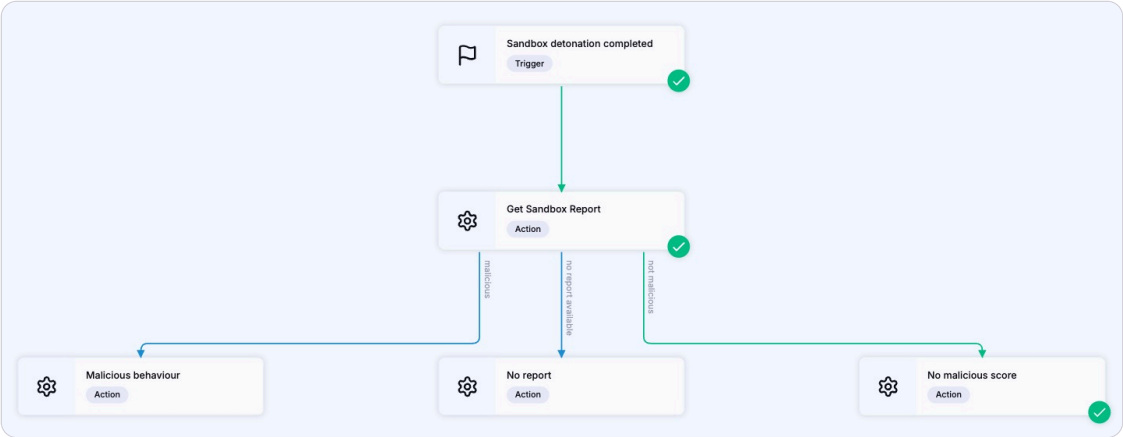

Figure 6: The trigger-driven template, started by itself when the detonation finished.

If something looks wrong. The first poll after you turn it on emits nothing at all. That is intended: it records the detonations that already exist rather than replaying your history into the playbook, and starts from the next new one. It is not broken, it is being quiet on purpose.

Triage a reported phishing email with PolySwarm

The job. "A user reported a phishing mail. I need the indicators out of it and checked, and I would rather not read the whole thing myself."

When to use it. On reported-phishing alerts. Pulling the indicators out of the message runs inside Sekoia and never calls PolySwarm, so the body of a customer's email does not leave their environment. Only the extracted URL is looked up.

When not to. It checks one URL from the message. For a mail carrying several indicators worth checking, use it as a starting point and widen it.

Search the catalog for: Triage a reported phishing email with PolySwarm.

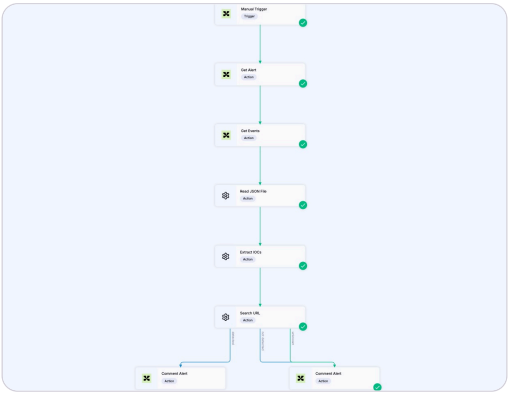

Figure 7: The phishing template, with the indicators pulled out of the message body.

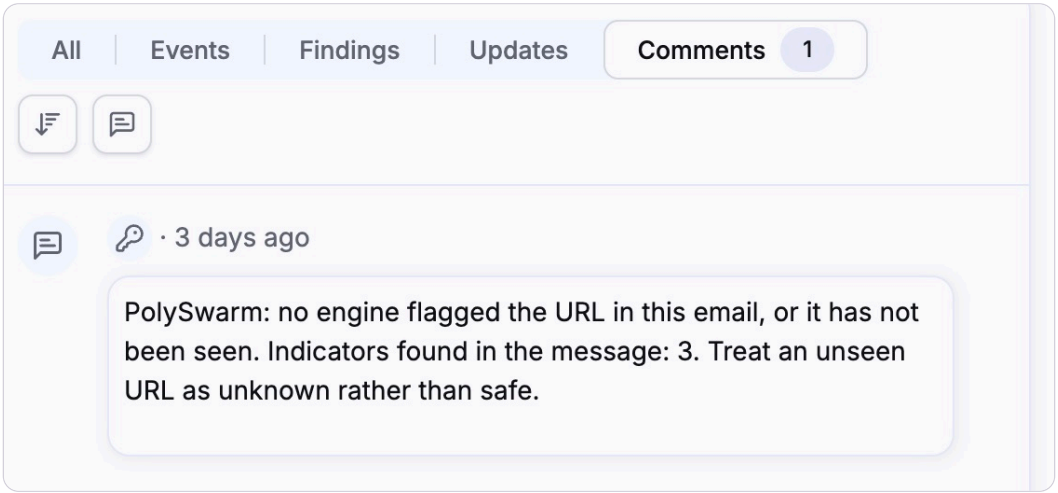

Figure 8: The comment, reporting how many indicators were found and what the lookup said.

If something looks wrong. An empty result usually means the message body was not where the template looked for it. Mail formats vary, and that field path is the first thing to adjust.

Pivot from an indicator with PolySwarm

The job. “I have a domain and no sample. What has been seen touching this?”

When to use it. When you have infrastructure rather than a file: a domain from an alert, and a question about what connects to it. Read only, so it costs no scanning quota.

When not to. It answers with artifacts associated with the indicator, not with a verdict on the indicator itself. For that, use the enrichment template.

Search the catalog for: Pivot from an indicator with PolySwarm.

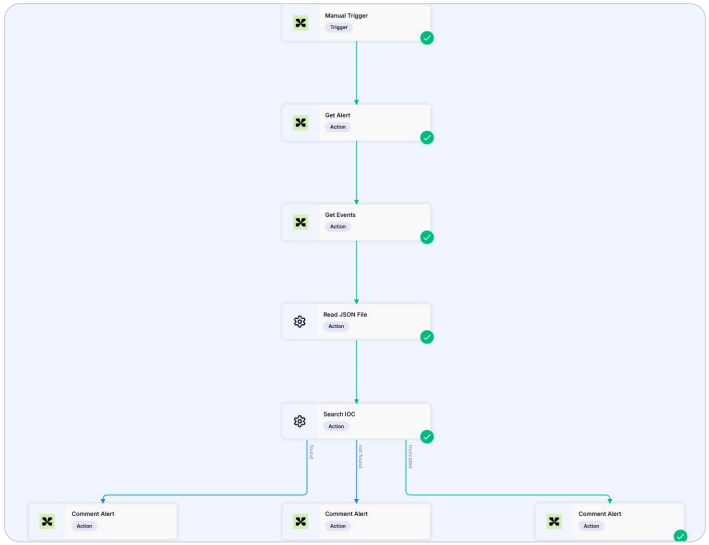

Figure 9: The pivot template, which took the truncated branch because more artifacts exist than the limit returned.



Figure 10: The comment, saying how many artifacts came back and that there are more.

If something looks wrong. A truncated result is not an error, it means more exist than the limit returned. Raise the limit to see the rest.

Check several hashes at once with PolySwarm

The job. “This alert has twenty hashes on it and I am not looking them up one at a time.”

When to use it. On alerts carrying many files. It looks them all up in one step, and duplicates are charged once.

When not to. Be aware this is a loop rather than a batch: fifty hashes cost fifty requests. On a very large list that is slow as well as expensive.

Search the catalog for: Check several hashes at once with PolySwarm.

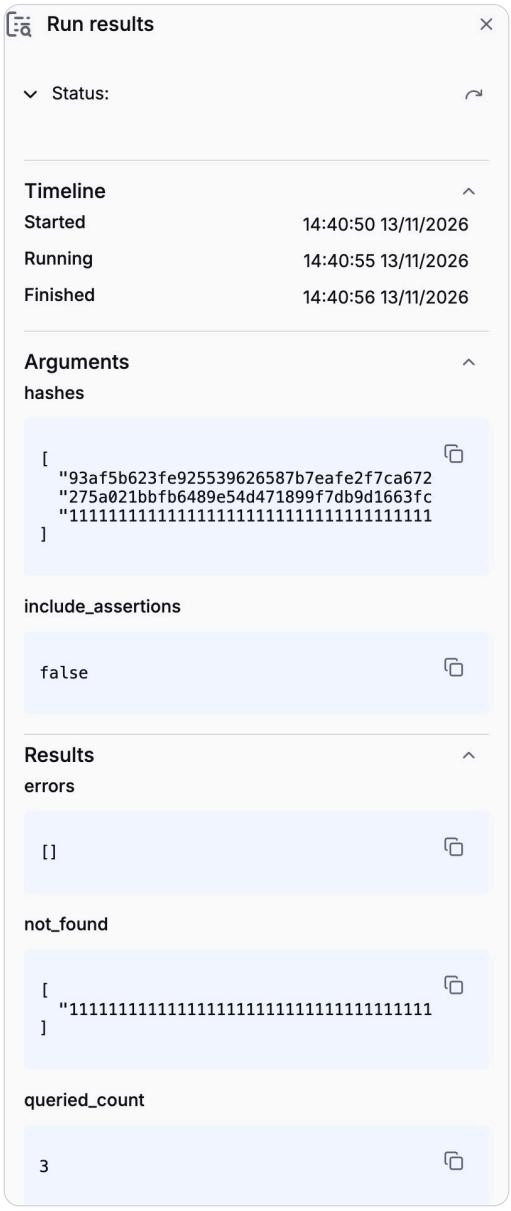

Figure 11: The result. The not found list holds only the hash that does not exist, so the mixed case is visible.

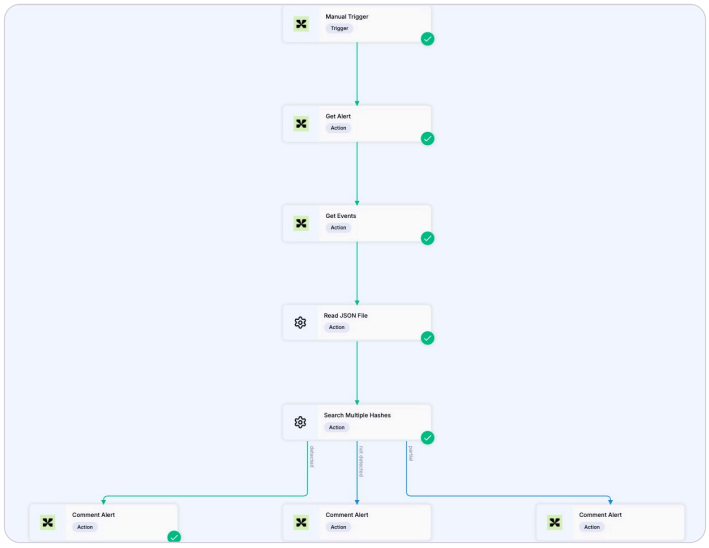

Figure 12: The bulk template after looking up three hashes in one run.



Figure 13: The comment, reporting how many were looked up and which were not found.

If something looks wrong, and this one matters. A run that stops early reports `partial` and returns the rows it already has. **Partial is an incomplete answer, never a clean one.** The hashes it never reached have no verdict at all, and treating the run as "nothing found" would be wrong. The template comments this explicitly so it is visible on the alert rather than buried.

Support

→ Email customersuccess@polyswarm.io.

→ Install, configure, the individual actions, quota and data handling are in PolySwarm for Sekoia User Guide.