

Security Bus and High Velocity Scanner PoC

October 19, 2026

Contents

1 Roles

- Security Analyst
- Compliance Analyst
- Developer
- Development Lead
- Product Owner
- Internal Security Researcher
- Incident Responder
- System Administrator

2 User stories

- As a Developer, I want to run SCA offline against my local repos so that I can know if any of the third party software I'm using to build my applications (sources, or applications) are vulnerable, and so that I can fix security issues identified during build SCA.
- As a Developer Lead, I want to run SCA during the build process so that I can block vulnerable software from being deployed.

- As a Developer Lead, I want to run override SCA blocking during the build process so that I can deploy critical fixes that may still register as security issues.
- As a Product Owner, I want to be able to view detected vulnerabilities so that I can understand the security profile, tech debt, and overhead of products I own.
- As a Product Owner, I want to be able to view detected vulnerabilities that have delayed fixes so that I can make sure issues get fixed.
- As a Security Analyst, I want to perform SCA for all software and operating systems (root file systems, docker images, git repos) I'm responsible for, so that I can identify emergent vulnerabilities or vulnerabilities that have slipped through build SCA.
- As Security Analyst, I want to rapidly create and deploy sensors so that I can identify new vulnerabilities for which there is not yet another tool (log4j). As an Internal Security Researcher, I want to be able to rapidly test, build, and deploy sensors that detect arbitrary signals so that I can perform experiments related to identifying vulnerabilities at scale
- As a Compliance Officer, I want to be able to produce compliance artifacts (SBOM, SCA run reports, vulnerability scanning reports) on demand so that I can demonstrate regulatory compliance (such as EU CRA)
- As a Product Owner, I want to be able to correlate between vulnerabilities found in a library, image, or repository and a product I own or product component, so that I can make sure issues are handled and handled by the right team.
- As any role, I want to be able to write alerting plugins so that I can be alerted about findings in my platform of choice (jira, slack, matrix, etc). I also want to be able to trigger other actions based on signals, so that I can automate actions I'd normally take.
- As a Security Analyst, I want to be able to look at the current security state of any system so that I can quantify the risk inherent in the attack surface.
- As a System Administrator, I'd like to compress recorded data so that my systems don't run out of space. I know data is valuable, so I don't

want to delete it. However, I do want to be able to prune old data (lossy compression).

- As an Incident Responder, I want to look back in time to find out when a vulnerability was introduced to the system, when it was detected, and what signals we have of exploitation so that this information can be used to reduce the risk of future issues. The value of granular data will decrease as time goes on. While I would like to know about the minute by minute of a response, I probably only need to know about the day (or possibly month) that a vulnerability was first introduced or detected.
- As a Security Analyst, I want to identify systems with similar vulnerabilities so that I can respond rapidly to other risk if one instance is exploited.
- As a Security Analyst or System Administrator, I want to be able to determine if actions are run individually or in batches so that I can tune the system to be more resource efficient.

3 similar tooling

- Wazuh (XDM/SIEM)
- scancode.io
- OWASP Dependancy-check
- Aqua Trivy
- nmap

4 Objectives

- SAST/DAST build/integration testing
- Parameter scanning

5 Immediate Objective

I am trying to identify vulnerabilities in <https://recipes.coopcloud.tech/>, trace image vulnerabilities to recipes, inform developers (via a git comment,

or pull request that upgrades vulnerable images to not-vulnerable), and generate data that can be used to create announcements. A recipe is a collection of docker images.

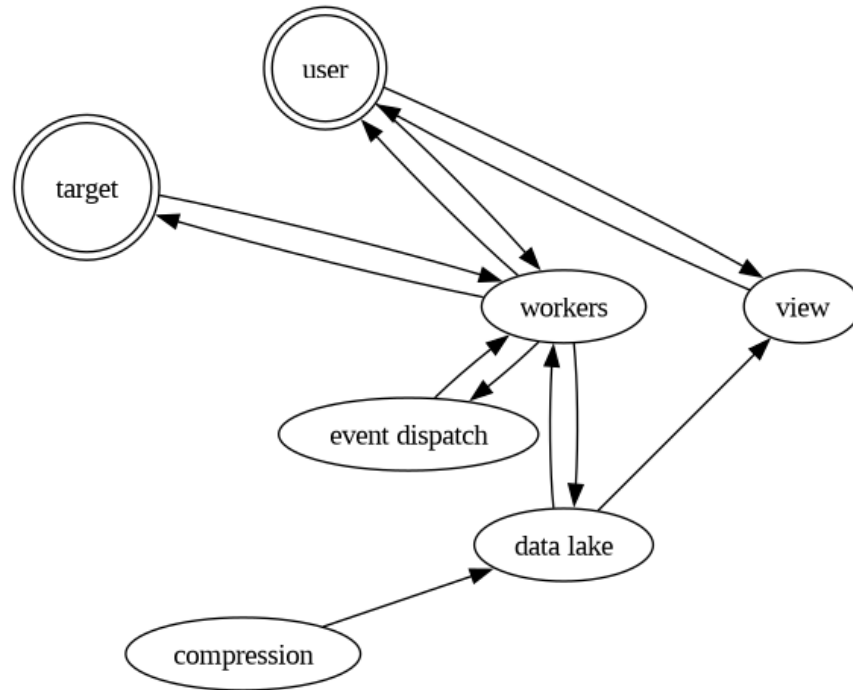
Scancode.io is slow. I don't need most of its functionality. It combines presentation with execution. It has crazy requirements. It's just a big 'ol mess. I want to build something small that I can run locally, but that I can also scale horizontally when I deploy to prod.

I see no reason I shouldn't be able to take the same architecture I'd use for recipe scanning and turn it around for custom vuln scanning or any number of other things.

6 Design Considerations

- Worker flows should be composable from workflow components simply by subscribing workers to different queues.
- The framework abstracts pub/sub complexity so user just writes functions or classes, input and output are handled by a wrapper.
- Improvements should be small and simple, only requiring knowledge about how to manage one behavior, one input, and not needing to know anything about any other part of the data model.
- It should scale horizontally easily, but be usable as a single command line application

7 Draft Design Notes



7.1 MVP Recipe Scanning Flow

1. User feeds Co-op Cloud Recipe in to system via a CLI that wraps a "Co-op Cloud Scanner" worker.
2. "Co-op Cloud Recipe" event fires.
3. "Recipe Parser Worker" subscribed to "Co-op Cloud Recipe" fires
 - (a) worker parses recipe into images
 - (b) worker saves recipe file into datalake (raw filesystem)
 - (c) worker publishes "Co-op Cloud Recipe Correlation Event"
 - (d) worker deduplicates images and publishes unique images to the "Scan Image" queue
 - (e) writes an event list to the data lake, triggers "Co-op Cloud Recipe Correlation" worker

4. Event Dispatch launches "Co-op Cloud Recipe Correlator Worker" (if not already running), which subs to "Image Vulnerability Event" and "Vulnerability Mapping"
5. "Scan Image Initiation" worker fires
 - (a) worker checks data lake to see if there is an existing SBOM for the image.
 - i. if not, it publishes the image url and to the "Docker SBOM Builder" topic
 - ii. if so, it publishes the SBOM uri and to the "Vulnerability Finder" topic
 - (b)
6. "Docker SBOM Builder" worker downloads image from docker hub, extracts, builds SBOM, writes to the data lake, deletes image so we don't run out of space (there are 2k images we're scanning here, so we have to do that... maybe we could put this in a ramdisk), publishes the SBOM uri to "Vulnerability Finder" topic
7. "Vulnerability Finder" worker loads SBOM, uses various methods to search for vulnerabilities, publishes unique vulnerabilities to "Vulnerability" topic and vuln+SBOM uri to the "Vulnerability Mapping" topic
8. "EPSS", "GhostCVE" and other enrichment workers worker subscribe to "Vulnerability" topic, writing their enrichment to the data lake
9. "Vulnerability Mapping" worker consumes from the "Vulnerability Mapping" until TIME or COUNT, batch updates vulm mapping in the data lake, pubs to "Vulnerability Mapping Complete"
10. "Co-op Cloud Recipe Correlation" worker consumes "Vulnerability Mapping Complete" in batches, mapping vulns back to Recipe, updating the data lake.
11. CLI worker monitors all events waiting for all tasks to be complete. When all tasks are either completed or failed, it writes a final report to the data lake and alerts the user that the run is complete.

8 technical challenges

- race conditions in the datalake
 - solution: simple file locking and batching for MVP
- knowing when a worker is taking a job so workers don't duplicate tasks, but also knowing when jobs are dropped
 - solution: worker receives all messages since last run, finds unspoken for tasks, writes to a claim q with a worker ID and timestamp, reads from the claim q a few milliseconds later to see if anyone else has claimed it. If no, then starts. If yes, then checks the timestamp. Whichever has the lowest timestamp wins. If both are equal, random jitters and tries again until there's a clear winner. Should be unlikely if timestamp has high enough resolution.
 - solution: DLQ monitored by event engine, try again. Event engine should monitor for failures. (This could just be another worker but would need a worker to monitor for recursive DLQ failure and trigger override to alert user and terminate the problem.)