Root Cause Analysis

Reporting data incident, 30th November 2021

Date of incident

The incident ran from 30th of November to 17th of December, from initial incident to resolution including backfilling of missing data.

Summary of incident

Customer impact

Elements of Snyk's reporting functionality were showing incorrect or inconsistent data, specifically in the reporting API (https://snyk.docs.apiary.io/#reference/reporting-api), any API call starting snyk.io/api/v1/reporting, and in the Summary and Issues tabs of our reports area in the UI. During the incident, for many customers the freshness of issues was impacted, dips were visible in the issues over time graph and exposure window graph, and the new issues, total issues and fixed issues total counts were inaccurate.

- Between the 30th of November and 6th of December, partial data was being returned in the snyk.io/api/v1/reporting API, reporting summary, and issues tabs.
- Between 6th of December and 17th of December, current data was returned in the API and UI, but the new issues, total issues count and fixed issues counts were inaccurate. Historic data from 30th of November to 6th of December remained incorrect.
- From 17th of December onwards, both current and historic data was correct.
- Services outside of reporting, including new issue notifications and the issues shown within projects including the project issues API, were not affected.

Technical summary

On the 30th of November, the extraction service used in reporting started running very slowly. This was due to purge jobs failing for a couple of days, leading to a significant increase in the data being processed. Extraction frequency is normally 90 minutes, but during the incident, extraction duration exceeded 90 minutes, leading to the data in the reporting database not ‘catching up’ with the current state.

During the incident, as the extraction duration exceeded frequency, incomplete data was being analysed. This caused the current issue counts to be lower than they should have been. Snyk’s reporting service performs a set of analysis, including comparing an issue that was there previously and checking whether it is still there, to determine whether it has been fixed. So, when the data did fully load, the current issues count returned to normal, but the counts for fixed issues and new issues had been incorrectly incremented, so were still inaccurate. The full and partial data also led to visible “data dips” on the issues over time graph. The exposure window graph also showed an inaccurate spike in new issues.

Incident root cause

A third-party service being used for extraction of issues data had intermittent failures, causing it to fall behind. The large size of one of Snyk’s tables made catching up from the lag a significant job. Implementing an alternative solution that contained the historical data was not something that could be achieved quickly. In parallel, an unexpected high load on the system exacerbated the slowness of the re-sync, and hindered the ability to run investigations and experiments for the fix on the system.

Incident timeline

Date - Time (UTC)	Action/Investigation.
30 Nov, 13:00	Incident was created due to data freshness being at 8 hours (normally 3 hours).
30 Nov, 14:00	Identified that ETLs were running for 4-5 hours, and vendor extraction times >2.5 hours. Ticket opened with extraction vendor.
30 Nov, 14:30	Purge jobs identified as failing causing significant increase in the volume of data being extracted. Fix to the purge jobs deployed.
1 Dec, 08:00	ETLs looking healthy, back at <90 minutes.
1 Dec, 14:00	ETLs remained healthy, data extraction (and thus freshness) improving.
2 Dec, 13:00	Data extraction behind again at 4-6 hours. Not failing, but also not catching up.
3 Dec, 09:00	New thread of action started to swap out the vendor extraction service for an alternative vendor/bespoke option. From here forward, the vendor extraction service is listed as "primary vendor" for purposes of clarity.
4 Dec, 10:00	Working with the primary vendor on strategies to fix the data extraction service, which would normally take ~20 minutes, but had been running on the same dataset for ~30 hours. Investigation into whether significant differences in data flow and levels on Snyk's side could have triggered the failures. Validated nothing out of the normal.
5 Dec, 04:00	Extractions are still running, but the primary vendor dashboard shows 30+ hours.
5 Dec, 14:00	The team working on an alternative vendor solution starts the sync of historic data in preparation for a possible switch if the

	primary vendor issue is still not resolved.
6 Dec, 15:30	The historic sync in the alternative vendor’s service is completed and tested. The reporting service is switched over to use this. Freshness is measured at under 3 hours, but fluctuating. The primary vendor service is still running, but from here forward, not in production. Status page is updated to indicate that the current data is now correct, although freshness is not guaranteed at <3 hours. Now data is current, a team is spun up to backfill the historical data.
10 Dec	A new, critical, wide spread vulnerability is disclosed (Log4Shell). A large number of new and existing customers begin running tests and importing significant numbers of new/additional projects. Snyk’s production system and database comes under significant and unplanned load. In order to ensure that production remains up and customers are able to identify and fix the Log4Shell issue in their projects, the team has to pause the large-scale queries that were running against the production database.
13 Dec, 14:00	Snyk’s production system is running under significant load during the first full ‘work day’ after the Log4Shell disclosure. The reporting service manages to retain freshness of between 2 and 5 hours, despite very large volumes of data being extracted.
16 Dec, 21:00	Data freshness on production is showing as close to 7 hours.
17 Dec, 11:00	The backfill of historical data from November 30th to December 6th is complete. All historical data has been backfilled for the duration of the incident.
17 Dec, 16:30	Production freshness is currently 3 hours. The incident is closed. A second Log4Shell wave causes fluctuations in reporting freshness, so freshness continues to be reported as 9 hours to set expectations, although it is anticipated to remain within the previous 3 hour window going forward. Work on stabilizing reporting continues.

Resolutions implemented, and actions taken to prevent a recurrence

- The reporting extraction service has been moved to an alternative vendor that has maintained more consistent levels of freshness.
- A significant reduction in table size was made to ensure more headroom in the short term, and a higher likelihood of any vendor being able to support extraction throughput.
- Additional monitoring has been implemented to increase visibility of issues with reporting's services.
- The "next generation" reporting infrastructure work that was already in progress has been accelerated. This is designed for greater scale, as well as increased flexibility and additional reporting capabilities.
- Work is continuing to reduce the table size for reporting, to make it more manageable and provide greater headroom for further growth.
- The team is investigating showing a flag on the reporting page to indicate current data freshness.
- The current database is being upgraded to a version that supports the vendor's recommended solution for Snyk's use case. This may provide more robustness in the extraction process, and reduce fluctuations in freshness caused by intermittent connector failures and recovery.