

Measuring what happens after an alert

Olamilekan Alaga / Product & On-chain Analyst | KOL

A bot-only case study of token monitoring, Telegram alerting and outcome measurement. Main alert period: 29 June to 15 September 2026. Results snapshot: 15 September 2026, 01:50 UTC.

11,142	11,308	166
Unique alerted tokens	Primary alert rows	Repeat primary rows

The problem I worked on

Early token markets produce more activity than someone can inspect manually. An alert is only useful if its starting point is recorded and its later outcomes can be measured. This project turns alerts into a research dataset that can be reviewed and improved.

What the supplied project records describe

A Python collector, SQLite storage, Telegram alerts and milestone updates, with deployment on Railway. This review checks the exported results; it does not independently inspect the live service or the source database.

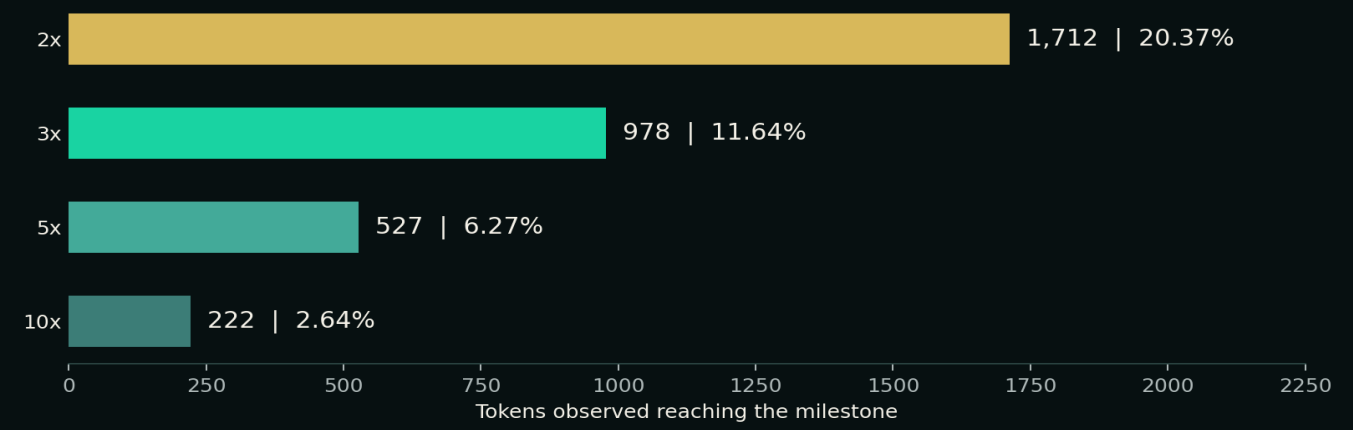
What the results show

The supplied 24-hour cohort contains 8,403 tokens. Of those, 1,712 were recorded reaching twice their baseline market cap. This is an observed alert outcome, not a realised trading return or evidence of performance above a benchmark.

Scope: Aladdin Alert Bot only. The terminal and blockchain LLM are separate projects and are not used as evidence in this report.

What the bot recorded within 24 hours

Recomputed from the supplied token-level export using its original eligibility flags. Thresholds overlap: a token reaching 10x is also included in the lower thresholds.



Milestone	Hits / 8,403	Rate	Median time*
2x	1,712	20.37%	14.0 min
3x	978	11.64%	23.8 min
5x	527	6.27%	41.9 min
10x	222	2.64%	72.7 min

*Median time is among tokens that hit that milestone, using recorded snapshots. It is not the waiting time for an arbitrary alert.

The remaining 6,691 cohort tokens had no observed 2x hit within 24 hours. This does not establish that all lost money; no executed trades are included.

Baseline is market cap, not a filled purchase price. Three tokens in this cohort use a later-snapshot fallback; four have no snapshots inside 24 hours despite being marked eligible. These original cohort figures are reproduced, not silently corrected. See page 5.

Why the seven-day count is smaller

The supplied analysis includes a token only when its last recorded snapshot extends to the end of the chosen window. Fewer tokens satisfy seven days than 24 hours. This rule does not guarantee continuous observation inside either window.

Window	Included	Incomplete / missing	Too recent
24 hours	8,403	2,648	91
7 days	5,862	4,565	715

Compare the same tokens to compare time windows

All 5,862 seven-day-eligible tokens also qualify for 24 hours. Holding that population fixed removes the changing-denominator problem:

Milestone	Within 24h	Within 7d	Additional hits
2x	1,160	1,176	16
3x	656	674	18
5x	346	359	13
10x	145	151	6

Across all available tracking

Among all 11,142 tokens, the export records 2,253 reaching 2x, 1,287 reaching 3x, 682 reaching 5x and 275 reaching 10x. Each token has a different follow-up length, so these totals should not be presented as a fixed-window success rate.

Selection matters: 510 additional tokens have an observed 2x hit within 24 hours but are excluded from the original 24-hour denominator because follow-up ended early or the token was too recent. Excluded tokens are not automatically failures.

The research dataset behind the bot

The export manifest describes the consolidated research database. These are reported export properties; the SQLite database and actual Parquet tables were not included in this ZIP.

Measure	Reported export value
SQLite size	13,887,336,448 bytes / 12.93 GiB
Exported tables	42
Rows across all tables	10,467,121
Compressed Parquet size	1.77 GiB
Row-count checks	All 42 table counts match in manifest

What those numbers mean

The row total includes snapshots, payload audits, schedules and synchronisation records. It is not a count of unique tokens, users or independent signals. Row-count agreement checks export completeness at one level; it does not prove every cell or model result is correct.

Alert records are different from update messages

The supplied summary reports 20,788 survivor milestone/reply rows and 5,322 entry-outcome milestone/reply rows. These are event counts, not additional unique successful tokens. The earlier legacy table is separate: 240 alert rows for 235 tokens and 360 milestone rows.

How this supports my analyst portfolio

The case study connects a user problem to a measurable product workflow: capture an alert, preserve its baseline, follow outcomes, deduplicate tokens and inspect tracking gaps before interpreting results. Those activities demonstrate analytical work beyond simply sending notifications.

ML research tables appear in the manifest, but this package does not establish a trained model's quality or its use in live alert decisions. No model-performance claim is made here.

What is verified, and what still needs evidence

Checks completed on the supplied files

Confirmed 11,142 unique token rows, first/latest alert dates, all 12 horizon-by-threshold hit counts and denominators, and median times to hit. Reproduced threshold flags from the exported maximum multiples. These checks validate internal consistency of the derived export, not the raw snapshot reconstruction.

Two corrections to the original description

Seven tokens use `first_post_alert_snapshot_fallback` rather than the alert-time market cap. Of these, three qualify for 24 hours and one for seven days. Four other 24-hour-eligible tokens have zero observations inside that window. “Full coverage” is therefore too strong a label.

A transparent sensitivity check excluding both fallback baselines and zero-in-window observations leaves 8,396 tokens, with 1,711 observed 2x hits (20.38%). This small adjustment does not solve unknown gaps between snapshots or follow-up selection bias.

Before claiming a trading advantage

Audit raw snapshot timestamps and market-cap fields, explain why tracking stops, and compare alerts against a suitable contemporaneous token population. Account for liquidity, execution, fees and slippage before estimating trade returns. Drawdown figures are snapshot-based and conditional on milestone hits.

Source files and missing assets

Sources: `telegram_alert_token_outcomes.csv`; `milestone_summary_by_horizon.csv`; `bot_alert_results_manifest.json`; `BOT_ALERT_RESULTS_BREAKDOWN.md`; `parquet_export_manifest.json`; `PARQUET_EXPORT_REPORT.md`. Project implementation descriptions come from the supplied handoff and were not re-audited against code.

No real Telegram screenshot or public bot URL was included. No terminal images are substituted. A real alert screenshot can be added later; the charts here are generated from the exported results.