

I ran this audit on my own trading bot first

Case study 01 audited a public template with nothing at stake. This one is different: a real algo-trading system wired to Interactive Brokers, currently in paper-trading validation ahead of a live decision. Same audit process, but this time it found bugs that were actually costing money.

SUBJECT	BROKER	UNIVERSE
Equity swing-trading bot (unnamed, active project)	Interactive Brokers, IB API	AAL, F, SOFI
STAGE	STYLE	
Paper trading — pre-live validation	Equity swing, ATR-based stops	

On 2026-07-22 the bot's live win rate came in at 11.1% against a backtested 45.9% on the same signal engine — the kind of gap most people blame on "the market changed." Instead of accepting that story, the fix was to re-run the exact same signals through three execution models, changing nothing else: idealized fills, then realistic fills (real cancellation rate, real slippage), then realistic fills plus the commission actually paid.

IDEALIZED BACKTEST
INSTANT FILL, NO FEES

0.254

Expectancy per trade, in R-multiples

REALISTIC EXECUTION
29% ENTRY CANCELS, REAL
SLIPPAGE

0.193

Barely moved — execution noise alone isn't the gap

+ ACTUAL COMMISSION
ROUND-TRIP, ~2% OF
NOTIONAL

−1.366

This is where the edge disappears

The middle number is the tell: order cancellations and fill slippage — the things people usually suspect first — cost almost nothing here. The real damage comes from a stop-loss sized to price volatility (ATR) getting compared against a commission sized to the full notional value of the trade. Narrow stop, percentage fee: the two were never compatible, and no amount of signal-tuning fixes it.

* "R" = R-multiple, a standard risk-management unit (1R = the amount risked on a single trade, i.e. the stop-loss distance). Not a currency.

CRITICAL

Protective orders expired at the end of the trading day

found 2026-07-11 · order routing layer

The bracket order that places the stop-loss and take-profit legs didn't set an explicit time-in-force, so the broker defaulted both legs to day-only. Any position still open at the close lost its protection silently — discovered when a Friday session ended with open positions and zero resting orders behind them. Fix: both legs explicit good-till-canceled.

CRITICAL

Stop-loss and take-profit were priced off the signal, not the fill

live case, 2026-07-10

On a limited entry, the bracket's protective levels were locked in before the order actually filled. If price moved during the wait for a fill — which limit orders by definition allow — the real risk/reward on the position could end up nothing like what the signal intended:

Signal price	\$16.87
Stop / target as planned	\$16.77 / \$17.06
Planned risk / reward	\$0.10 / \$0.19
Actual fill	\$17.03
Real risk / reward	\$0.26 / \$0.03

A trade planned at roughly 2:1 reward-to-risk executed at closer to 9:1 *against*. Fix: hold the bracket's stop/target until the entry is confirmed filled, then reapply the same offsets to the actual fill price.

HIGH

A hardcoded exchange silently disabled risk checks for one symbol

found 2026-07-22 · data feed layer

The price-quoting code assumed every symbol traded on Nasdaq — true for two of the three names in the universe, not for the one listed on the NYSE. Every price lookup for that symbol failed with an unhandled contract error, which meant the anti-slippage check silently saw "no current price" and always passed. Every entry on that symbol was priced off a stale signal-bar close instead of a live quote — for weeks, invisibly. This was one contributor to the 45.9%-vs-11.1% backtest/live gap — found before the deeper commission-driven explanation surfaced the next day.

03 — WHY A BACKTEST DOESN'T CATCH THESE

THE PATTERN ACROSS ALL THREE

- All three assume the happy path: the order always fills at the intended price, the process never restarts, every symbol behaves like the one it was tested on first.
- None of them are signal problems. Entry logic, indicators, filters — untouched. Every fix was to order management, execution timing, or data plumbing.
- Every one was silent. No exception surfaced to a dashboard; each degraded a specific position or symbol while everything else kept reporting normally.
- A backtest that replays historical bars through the signal engine will never reproduce a day-only order default, a fill-timing race, or a hardcoded exchange string — because a backtest doesn't route real orders.

This bot is currently in paper trading — a validation stage prior to any live-capital decision, run against real Interactive Brokers market data and order routing rather than a spreadsheet simulation. All three bugs above are fixed as of their respective dates; the project's name, exact strategy parameters (entry thresholds, stop-distance sizing), and account figures are withheld as this is an active, unpublished trading system with no live track record yet — not a product for sale.

Same audit, for your bot

I ran this process on my own system before offering it to anyone else — these three bugs and the honest-backtest reversal above are what it actually found. If your bot has real capital behind it, or you're deciding whether to buy someone else's, the same review applies.

RISK REVIEW

Order time-in-force, fill-timing races, what happens when the bot restarts mid-position.

HONEST BACKTEST

Re-run with realistic execution and actual commission — not the idealized number.

DATA-PATH AUDIT

Silent failures in quoting/contract logic that quietly disable risk checks for specific symbols.

CODE REVIEW

Logic bugs in entry/exit/stop handling that only show up once a live fill disagrees with the signal.

Written report, turnaround under a week. honest.audit48@gmail.com

Case study 02 of an independent trading-bot audit practice — see case study 01 for a public-template audit with the same methodology.