FORESHOCK

Foreshock's First Backtest: What Three Verifiable Signals Flag in DeFi Exploit History, and What They Don't

Every number below is traceable to a named source in Foreshock's own records, cited as it comes up, so it can be checked without taking Foreshock's word for it.

This is not financial, legal, or insurance advice, and nothing below is a guarantee of safety or risk for any protocol, named or not. The original backtest below ran under methodology version 0.1.0; later sections describe what changed through version 0.6.0, with every score labeled by which version produced it. Read every number here as a research finding, not a live risk assessment. See the note at the end of this document for what's changed since this was written, including the live methodology version today.

Why Foreshock exists

DeFi insurance is a real, growing market. Billions of dollars in on-chain assets are already covered by mutuals and cover protocols, and the demand side keeps expanding as more institutions hold crypto on-chain. Industry reports, independent analysts, and the insurers now entering the space all acknowledge the same open problem: accurate risk pricing hasn't been solved. Cover today is priced largely by supply and demand, not by an independent, reliable read on which protocols are actually risky and why.

Foreshock is being built as that independent read. It's a risk-intelligence system for DeFi protocols: it maintains its own cross-referenced ledger of historical exploits, pulled from multiple public trackers so no single source's gaps become Foreshock's gaps, and it scores protocols on risk signals that are mechanically verifiable, checkable against public data, not asserted. This piece is the first test of whether that scoring actually says anything true.

What Foreshock did, at the time of this first backtest

At the time of this first backtest, Foreshock scored protocols on three mechanically verifiable parameters: **incident history** (has this exact protocol been hacked before, and when), **TVL trajectory** (TVL is total value locked, meaning a protocol's size in dollars, and whether it's moving fast), and **protocol age**. Six further categories were part of the full scoring rubric and under active development: audit history, code characteristics, dependency risk, team factors, bug bounty programs, and governance attack surface. None of them could be reconstructed for a historical date in this backtest, so all six sat at a fixed neutral value in every result below. This isn't a guess. It's a placeholder for "not yet built," stated plainly rather than folded into the numbers. (**What's changed since:** audit history joined the live rubric for backtests 2 onward, and four more categories are now live for current-day scoring, though structurally never backtestable. See "What's changed since," below.)

That means this backtest doesn't test Foreshock's full rubric, and it can't show detection via code analysis, audit gaps, or governance red flags. Those categories weren't live for any record here. What it tests is narrower and answerable: do the three parameters Foreshock can already check, on their own, actually separate protocols that get hacked from protocols that don't?

How the test works

Foreshock rewinds the clock to the day before a real historical hack, scores the protocol using only data that was actually knowable that day, and compares the result against a control group: a matched set of protocols, similar in size, age, and category, that never got hacked in the same window. Point-in-time discipline is the whole point: a score computed with hindsight isn't a backtest, it's a rationalization.

The overall picture

Foreshock's ledger holds 642 incidents, cross-referenced between DeFiLlama and Rekt.news (plus a small, explicitly partial set of Nexus Mutual claims data; see Sources, below). Of those, Foreshock scored 192 and excluded the rest for stated reasons: 309 had no DeFiLlama protocol ID to match against (DeFiLlama's own hack data frequently omits this, even for incidents it sourced itself), 4 had an ID matching nothing in DeFiLlama's protocol list, and 137 had a matching protocol but not enough historical data to reconstruct even these three parameters (the Ronin case study below is the clearest example of why).

Foreshock matched each of the 192 against a stratified control group of 374 never-hacked protocols, size- age- and category-matched as of the same historical date as each incident, never today's data. The two groups' scores:

	how many	lowest	typical (median)	average	top 10%	highest
Incidents	192	42.25	43.5	44.87	48.25	**57.5**
Controls	374	42.25	43.5	43.72	46.25	**48.25**

Typical scores are identical, which is expected, since six of nine categories are neutral for everyone in this test. The separation that exists is real but thin, and lives entirely at the top: incidents reach 57.5, controls never exceed 48.25.

Against a pre-committed threshold (control average plus two standard deviations, fixed from the control data alone before Foreshock scored a single incident; see Assumptions, below, for why the order matters): **precision 0.426** (of everything flagged, 42.6% was a real incident), **recall 0.271** (of all real incidents, 27.1% got flagged), **false-positive rate 0.187** (18.7% of healthy controls got flagged anyway). At this incident-to-control ratio, chance alone lands around 34% precision. 42.6% is real, modest lift, not a strong classifier, and this isn't being presented as one.

Why incident history carried the tail

Seventeen of the 192 scores land above 48.25, higher than any control ever recorded. All 17 are elevated for the same reason: the protocol had at least one prior recorded incident, which floors that category's score high regardless of anything else. TVL trajectory and protocol age shift the number in some cases, but neither alone ever reaches flagging level. Incident history did nearly all of the separating.

TVL velocity, meaning sudden inflows, the kind of spike that raises the incentive to attack, mattered less than expected. Across the tail, it barely moved the needle; all 17 elevated scores would look much the same without it.

So: is "already hacked once" a meaningful warning sign, or does anything happen twice about this often? Worth walking through plainly before the arithmetic.

The repeat rate. Restricting to incidents tied to a specific DeFiLlama protocol ID (333 of 642, the same coverage limit as the scoring above), 287 distinct protocols had at least one incident, and 39 had a second:

> 39 / 287 = **13.6%**

First comparison, and it's the wrong one, though it's the first thing an adversarial reader will try. Compare 13.6% against how often *any* protocol in the whole DeFiLlama-tracked universe (7 788 protocols) has *any* recorded incident at all:

> 287 / 7 788 = **3.7%**

13.6% against 3.7% is a real gap, roughly 3.7x, but this is still the wrong comparison, and it's worth being precise about why: it quietly swaps the question. "How often does a hacked protocol get hacked again" and "how often does a random protocol get hacked at all" are different events. This comparison doesn't erase the signal. It understates it, because it's answering a related but different question.

The comparison that actually answers it holds the event fixed (reaching two or more incidents) and only changes what's already known going in:

> 39 / 7 788 = **0.50%**

Against that, 13.6% is a **~27x** lift: same outcome, asked twice, once blind and once knowing the protocol had a prior incident.

A sharper objection, worth taking seriously rather than waving off. Hacks aren't spread evenly. A handful of protocols get targeted repeatedly, and most never get touched. So even with zero memory effect, even if a protocol's past had no bearing on its future at all, some protocols would still get unlucky twice, the way a fair coin sometimes lands heads three times running. How much of that 0.50% is ordinary bad luck, and how much is left over once it's subtracted out?

Modeling a world where every recorded incident lands on a protocol chosen completely at random, with no protocol more likely than any other and no memory of the past: how many protocols would end up hit twice by pure coincidence?

The answer depends on one detail that can't be fully resolved: how many incidents to charge to the universe, given that DeFiLlama's own data frequently doesn't say which

protocol a hack hit. So this is computed two ways.

Conservative: charge all 642 ledger incidents to the 7 788-protocol universe, average incidents per protocol $\lambda = 642 / 7\,788 \approx 0.0824$. Chance of any protocol getting hit twice or more, purely at random:

> $1 – [e^{(–0.0824)} × (1 + 0.0824)] \approx$ **0.32%**

Against that, 13.6% is a **~42x** excess.

Strict: only charge the 333 incidents attributable to a protocol ID, $\lambda = 333 / 7\,788 \approx 0.0428$:

> $1 – [e^{(–0.0428)} × (1 + 0.0428)] \approx$ **0.089%**

Against that, 13.6% is a **~153x** excess.

Neither end is "the" answer. The true figure depends on how many of the 309 unattributed incidents belong to protocols that already appear elsewhere in the linked set, which isn't determinable from the data at hand. Reporting a range rather than the number that looks best: **somewhere between roughly 40x and 150x above what pure chance predicts, depending on how incidents with missing protocol IDs are counted. Even the conservative end is a full order of magnitude above chance.**

One more check, so the choice of population isn't doing the work. Rerunning the simple version (repeat rate against population base rate) on the full, messier ledger (protocols matched by name where no DeFiLlama ID exists, not just the strict ID-linked set) gives a smaller lift, about 13x instead of 27x, but the lift is still there. Which population you count changes the exact multiple. It doesn't change the conclusion that a prior incident is a real, non-trivial warning sign.

That's the mechanism behind both case studies below.

Case study 1: Poly Network

Poly Network's first incident landed on August 10, 2021: $611 000 000 lost, one of the largest DeFi exploits on record, confirmed by both DeFiLlama and Rekt.news. Its second incident, the one Foreshock scored, came on July 2, 2023: a private key compromise, $5 000 000 lost, also confirmed by both trackers. Foreshock's reconstructed score the day before that second incident was 52.75, against a threshold of 42.25 for its group (large-size, established-age, Bridge protocols). Flagged.

Breakdown: incident history scored 85 of 100 (weight 0.15, for a prior incident of its own), TVL profile scored 35 (weight 0.10, TVL essentially flat, down 5%, nothing unusual), protocol age scored 30 (weight 0.05, 395 days old, past the point where youth counts against a protocol in this model). The other six categories sit at neutral 50, exactly 70% of total weight, unresearched for this date. The elevated score is arithmetically 100% attributable to incident history; TVL and age contributed nothing unusual.

That floor is durable, not just true on this one date. Once incident history locked in at 85 (weight 0.15) the day after the first incident, that component alone combined with the six neutral categories at 50 (0.70 of total weight) already summed to 47.75, computed as 0.15×85 plus 0.70×50. That total alone clears the 42.25 threshold, which means the reconstructed score stood above the flagging threshold for the entire stretch between the

two incidents, regardless of how TVL or age moved in between.

At the time this backtest ran, Foreshock's confidence accounting reported this as low confidence for two stated reasons: only 30% of category weight behind it was verified data, and separately, confidence accounting had no recorded backtest history for this specific group to draw on, a gap in what was wired up at the time, not a claim that no backtest existed. That gap wasn't rounded away to "medium" confidence because it would have read better next to a backtest result.

What this does and doesn't show. It doesn't show Foreshock spotted anything about Poly Network's code, audits, or bridge design that let it foresee the 2023 incident. It shows that one plain, checkable fact, a prior recorded incident on this protocol, was already in the data and correctly pushed the score up.

Case study 2: Radiant Capital

Radiant Capital's first incident was on January 2, 2024: $4 500 000, confirmed by both trackers. Its second incident, the one Foreshock scored, came on October 16, 2024: $53 000 000, an access control exploit, the largest dollar loss anywhere in the scored tail, confirmed by both trackers and one of the 18 pairs matched by hand during ledger construction, because DeFiLlama and Rekt.news named it too differently for automatic matching ("Radiant V2" vs. "Radiant Capital - Rekt II"). Foreshock's reconstructed score the day before that second incident was 52.75 against a threshold of 42.25 (mid-size, established-age, Lending). Flagged.

Same shape as Poly Network: the same six categories neutral at 50, incident history at 85 doing the work, TVL profile at 35 (down 21%, still inside the model's "stable" range), protocol age at 30 (574 days old, past the youth penalty). Same low confidence, same stated reasons.

Two different protocols: nine months between incidents for one, nearly two years for the other; one a bridge, one a lending market; different chains. Same single mechanism doing the work both times: each protocol's own prior incident, plus public TVL history, nothing else.

Case study 3: Ronin, the hack Foreshock couldn't score, and why that matters more than one it could

Ronin Network's incidents are, by dollar amount, among the largest in the ledger: $624 000 000 on March 23, 2022, then $12 000 000 on August 6, 2024. Both are confirmed by DeFiLlama and Rekt.news. **Foreshock couldn't score either one.**

The reason is specific and checkable: DeFiLlama's protocol record for this entity (ID 4440, slug `ronin-bridge`, category "Canonical Bridge") has no listing date at all. Not an old one, none. Without it, protocol age can't be computed for any historical date, which alone excludes both incidents from the matching process, regardless of a separately confirmed fact: this slug's own TVL history, under its current listing, only goes back to April 5, 2024, after the 2022 hack had already happened.

Worth being careful here: there's no evidence DeFiLlama did anything deliberate, and this piece isn't suggesting otherwise. What can be said is narrower, and matters more: **the**

hack itself isn't forgotten (two independent trackers still confirm it happened, in Foreshock's own merged ledger), but the protocol-level context needed to actually score it is no longer retrievable from the largest single data aggregator in the space. That can happen through relisting, restructuring, or ordinary data lifecycle churn after a major incident, with nobody hiding anything. A protocol's own worst moment can quietly become invisible in a downstream aggregator's current snapshot, for reasons that have nothing to do with intent. That's exactly the failure mode an independent, cross-referenced ledger exists to survive: one that keeps its own record instead of depending on any single upstream aggregator's current view. Ronin is the clearest example available of why that independence isn't optional.

Limitations

- **Foreshock scored 192 of 642 ledger incidents** and excluded the rest for stated, checkable reasons: no protocol ID, no matching protocol record, or not enough historical data to reconstruct even the three live parameters. It did not select a nicer-looking set.
- **Six of nine rubric categories were neutral for every record in this backtest**, by construction: audit history, code characteristics, dependency risk, team factors, bug bounty, and governance attack surface have no historical-dated source yet. This backtest tests whether the three live parameters, in practice mostly incident history, carry signal on their own. It does not test the full rubric's real-world performance.
- **Every backtest-1 score in this piece reports low confidence, by design: a historical fact about that run, not a claim about current live scoring.**
- **Precision (0.426), recall (0.271), and false-positive rate (0.187) are diagnostics over a partial rubric**, not a claim about live performance.
- **The control group has its own disclosed, unresolved bias:** control candidates can only be drawn from protocols DeFiLlama still tracks today. A protocol that was genuinely healthy at some past date but has since been delisted or abandoned for reasons unrelated to any hack is invisible to this control pool. The stratified matching does not correct for this.
- **Foreshock fixed the anomaly threshold before scoring a single incident**, computed from control data alone, specifically so it couldn't be tuned to catch more incidents after the fact.

Assumptions, stated plainly

- **Population for the base-rate comparison:** the headline figures (13.6% / 0.50% / ~27x) use only incidents tied to a specific DeFiLlama protocol ID, so both sides of every ratio are drawn from the same tracked universe. The same comparison on the full, unlinked-inclusive ledger gives a smaller but still-present lift (~13x); see "one more check," above.
- **Threshold pre-commitment:** Foreshock fixed and logged the anomaly threshold (control mean + 2 standard deviations, per bucket, with a pooled fallback for buckets with fewer than 5 controls) before comparing any incident's score against it. This is what keeps the threshold from becoming a second, hidden way to fit the results.
- **Control-group construction:** for each incident, controls are protocols matched on size band, age band, and category as of that incident's own historical date, never today's, and confirmed to have no recorded incident as of that date. A protocol that gets hacked later doesn't retroactively disqualify it as a control for an earlier date.
- **What "neutral" means:** a category scored neutral is not a midpoint guess about that protocol specifically. It's a fixed placeholder applied identically to every record where the underlying data doesn't exist yet, so absence of information never reads as evidence of safety.

What's changed since: backtests 2, 2.5, and 3

This piece was written after backtest 1. Three more runs have happened since, each adding one real capability to the live rubric and testing it the same disciplined way: design the rule from domain judgment first, write it down, version the methodology, and *only then* run the backtest, never the other way around.

Backtest 2 (methodology 0.2.0) added audit history, and immediately surfaced why "audited" is a dangerous word to score naively. A protocol can have a real, confirmed audit on record, or it can have only proof that a contest *happened*, a Code4rena or Sherlock repo that exists, with nobody having read what it found. Those are different facts, and scoring them the same would let "we don't know" quietly read as "checked, and it's fine." Foreshock's audit-history category enforces the distinction structurally: an audit record can't report findings without first claiming it actually knows them; records that are existence-only score a flat, neutral 50, mathematically indistinguishable from no audit at all. The real numbers bore this out starkly: of 566 backtest records, only 2 had audit data that actually moved the score, because most of the rest that had *some* audit record found were existence-only. Automated ingestion at conservative, exact-match precision mostly finds "a contest happened," not "here's what it found."

Backtest 2.5 (still 0.2.0, no rubric change) parsed real severity counts out of those same contest repos, closing part of that gap. 96 of 127 matched Code4rena/Sherlock audit records got real High/Medium counts instead of staying existence-only. But, checked directly rather than assumed, the existing 0.2.0 rubric didn't yet read those counts for anything beyond the same findings-known/not-known distinction, so this pass mostly converted existence-only records into "confirmed, findings known" records without yet changing what the severity numbers *meant* to the score.

Backtest 3 (methodology 0.3.0) is where severity counts started to matter: designed and version-bumped before a single result was checked. The rule: a protocol's most recent audit's High and Medium finding counts now shape its audit-history score, banded rather than scored as a raw count (zero findings scores best; a handful is unremarkable; a moderate and a high count step up in stated increments). Designed from how contest economics and vote delegation actually work, not from watching which threshold made the backtest look better. The design was written down before the backtest ran, along with a stated prediction: *"a small number of records shift, similar in scale to backtest 2.5's 3 incidents + 4 controls."* The actual result: **5 incidents and 5 controls moved**, the right order of magnitude; the exact count undershot rather than hit precisely, and it's reported as exactly that rather than rounded up to "confirmed."

A finding worth stating plainly, because it cuts against a naive reading of "more audits, better signal": zero of the 96 automated Code4rena/Sherlock findings records had zero High-or-Medium findings. Competitive audit contests, real ones, essentially always find *something*. The two "clean audit" scores that did move in backtest 3, Euler V1 and Radiant Capital (the same Radiant Capital named in the case study above), came from manually-researched, explicitly-confirmed-clean audits, not from the automated pipeline. (To be precise about which number is which, since the same incident now has three: the case study above cites Radiant's score under the original 0.1.0 methodology, 52.75. Under the later, audit-aware 0.3.0 methodology, that same incident's reconstructed score is a different number, 47.75, down from 49.75 under 0.2.0/backtest 2.5, because it's a different rubric being applied to the same historical facts, not a correction of the earlier figure. Scores from different methodology versions are never directly comparable, and every mention above says which version produced it.)

The metrics moved, but "close to a wash" is still the honest read, every time. Across backtest 1, 2, 2.5, and 3, in order: precision went 0.426 → 0.432 → 0.455 → 0.469; recall went 0.271 → 0.250 → 0.234 → 0.240; false-positive rate went 0.187 → 0.168 → 0.144 → 0.139. Precision and false-positive rate moved in one consistent direction across all four; recall dipped before ticking back up. At every single step, fewer than a dozen of 566 records actually changed. That is not a demonstrated trend at this sample size, and every comparison document behind these numbers says so explicitly rather than letting a favorable-looking arrow speak for itself.

What backtests 2, 2.5, and 3, together, actually demonstrate isn't a better score. It's that the discipline holds under real pressure to cut a corner. Every one of these rounds created a natural temptation: severity counts existed in the data and weren't yet being used. The obvious next move, done the ordinary way, is "try reading them, tune until separation improves, ship it." That is precisely what didn't happen. The rule was designed from domain reasoning, written down, version-bumped, and *only then* run against the backtest, with the predicted result stated in advance, checked against the actual result afterward, and the gap between them reported honestly rather than smoothed over. That sequencing is the actual product of these three rounds. The metric movement is a footnote to it, not the other way around.

What Foreshock scored at the time this was written

Live scoring is a different thing from backtesting, and the difference matters enough to state plainly rather than let the two blur together. A backtest can only use data that's reconstructable *as of a specific past date*: four of the rubric's nine categories qualify (incident history, TVL trajectory, protocol age, and, as of methodology 0.2.0, audit history). Four more categories were live at the time this was written but structurally can never enter a backtest, by design: code characteristics (admin key structure, upgradeability), dependency risk (oracle and bridge reliance), governance attack surface (token concentration, quorum), and bug bounty coverage all describe a protocol's state *right now*, not any historical date. Using today's value to score a 2022 incident would be exactly the kind of hindsight leak this whole project exists to prevent, so the code enforces it structurally, with a test proving that current-state data smuggled into a backtest reconstruction gets ignored, not silently accepted.

That left eight of nine categories with a real, live data path at the time. Only team factors (is the team doxxed, what's their track record) had no automated ingestion; it is researched manually per protocol, on request. Coverage on those eight was honestly pilot-scale, not universal: on-chain code-characteristics and governance data came from a small, matched pilot set of protocols with manually-verified core-contract or governance-token addresses (address resolution, finding the right contract to even ask about, has been the recurring bottleneck, not the on-chain reads themselves, which are free and effectively instant once an address is confirmed); bug bounty coverage came from a targeted, conservative lookup against one platform, not a bulk crawl.

Aave V1, scored live at the time this was written, was the concrete depth demo. Seven of nine categories carried real, verified data for it: 78% data completeness, among the highest of any protocol scored live at the time. One category told an honest-adjustment story worth walking through on its own. Aave V1's governance token's top 10 holders control 41.57% of supply by raw balance. Reported alone, that number would read as dangerous concentration. Adjusted for what those top holders actually *are* (a staking contract, the protocol's own ecosystem reserve, three exchange custody wallets, each excluded only on a positive, checkable tag match, never a guess), the real figure is 15.14%. Both numbers are kept, always, not just the flattering one: the raw figure stays visible specifically so the adjustment can be checked, not trusted on faith.

Sources

This backtest draws on two independent public trackers, DeFiLlama's hack database and Rekt.news's leaderboard, cross-referenced against each other, plus a small, explicitly partial file of Nexus Mutual claims data (one manually verified entry as of this writing; not a complete claims history). Foreshock's ledger architecture is source-agnostic by design: additional trackers can be added without changing how scoring works.

Methodology and how to check these numbers

Backtest 1, described above, ran under methodology version 0.1.0. Category weights were chosen by hand, written down before backtest 1 ran, and never fit or optimized against the incident and control labels shown in this piece. Fitting against these exact labels is precisely the shortcut a rubric like this could take, and precisely what this was built to avoid. Every rule added since (0.2.0's audit-existence distinction, 0.3.0's severity banding, 0.4.0's governance-holder exclusion taxonomy) followed the same discipline: designed and documented before being run against any result, never after.

Editor's note: the methodology has since advanced from 0.4.0 (the version live when this report was originally drafted) through 0.6.0 and, as of this publication, 0.7.0. Version 0.5.0 added a one-hop resolution path for contracts fronted by an OpenZeppelin ProxyAdmin and a new governance-executor admin-key classification; version 0.6.0 added an additive risk factor for oracle price sources that a sampling pass couldn't independently identify; version 0.7.0 added disclosed handling for protocols with a secondary governance veto mechanism (documented in full in `scoring/METHODOLOGY.md`*). None of these later changes have been re-run through a full backtest yet - the precision/recall figures above (0.426 through 0.469, ending at backtest 3 under methodology 0.3.0) describe an earlier rubric, not the one scoring protocols today. A fresh full backtest under the current methodology is on the roadmap; until it runs, the honest position is to stand behind the validated ~27x repeat-incident finding, which has held up across every methodology revision checked so far, and not restate older precision numbers as if they described today's live scoring.*

Every score referenced here is a probabilistic, backtested signal from an automated system. It is not financial, legal, or insurance advice, not a guarantee of safety or risk, and must not be the sole basis for a coverage, investment, or treasury decision. Confidence levels and methodology travel with every figure above. Verify independently before acting on any of it.