FORESHOCK

# Methodology, summarized

*Not financial, legal, or insurance advice. This document summarizes how Foreshock's scores are built. It is not the full internal rubric.*

Foreshock publishes an independent risk score for DeFi protocols, built from nine categories of evidence and delivered with the reasoning and confidence level attached. A score is a structured read of the risk signals present in public data. It is not a forecast, and it is not a safety rating.

## What each category examines

**Audit history.** Which firms reviewed the code and when, how many reviews exist, and whether any critical finding is recorded as unresolved. A review whose findings we have not actually read is tracked as exactly that, and can never be counted as a confirmed-clean audit.

**Historical incidents.** The protocol's own past exploits, plus incidents in the codebase it was forked from. Both carry forward permanently, independent of how large or established the protocol has since become.

**Dependency risk.** What the protocol relies on to function: bridges, oracles, and composability exposure to other protocols. Bridge dependence is treated as the most severe, grounded in our own ledger, where the largest-dollar incidents on record are bridge compromises.

**Code characteristics.** Whether the code is open source, whether contracts are upgradeable, what type of admin key controls them, and the multisig threshold behind that key. Who can change the contracts, and how easily.

**Governance attack surface.** Concentration of governance token supply among the largest holders, and the quorum required to pass a proposal. Together these describe how few parties would need to cooperate to control an outcome.

**TVL profile.** Deliberately measures instability, never size. Only the movement of total value locked over recent windows is read; absolute TVL is never an input, so large alone earns nothing.

**Team factors.** Whether the team is publicly identified, and its track record across prior protocols.

**Bug bounty coverage.** Whether a program exists, the maximum payout on offer, and the platform hosting it. A funded bounty is standing evidence that finding a bug is more rewarding than exploiting it.

**Protocol age.** Structured as a penalty for being new, with a floor, never a reward for being old. Once a protocol is past its early period the age contribution stops improving permanently, so a long-lived protocol never scores better on age than a merely established one.

## The principles behind the score

**Verifiable data, never guessed.** Every fact behind a score traces to a named, checkable source: a protocol's own docs or repository, a live on-chain read, or a labeled third-party dataset. Nothing is inferred to fill a gap. An address, an admin structure, or a dependency is either confirmed or it is not scored as confirmed.

**Missing data is never scored as safety.** A category that has not been researched is held at a neutral position rather than a low, safe-looking one. A protocol with several unresearched categories reads as genuinely unassessed, and can never be mistaken for one that was checked and came back clean.

**The rubric is fixed, not fitted.** Category weights and scoring rules are chosen by domain judgment and written down before any backtest runs against real historical incidents. Backtesting measures the rubric's accuracy; it never trains or tunes it. Fitting the rules to past data would let the model rediscover "big and old protocols survive" as a shortcut instead of measuring risk, partly because protocols that are badly exploited often do not stay big for long.

## Three guardrails against "big and old means safe"

Size and longevity are the two facts available for free about every protocol, which makes them the easiest thing for any risk model to accidentally reward. Three rules exist specifically to prevent that:

**Size is never safety.** The TVL category reads volatility only. A protocol holding billions gets no credit for that fact anywhere in the rubric.

**Age stops paying.** Age reduces risk contribution only while a protocol is genuinely new, then flattens permanently. A veteran protocol and a merely established one score identically here.

**History follows the code.** A protocol's own incidents, and incidents in the codebase it forked from, persist in the score no matter how large or old it later becomes. A big, veteran protocol built on previously exploited code stays flagged.

## Reading a score

**Confidence** states how much of the rubric is actually verified for that protocol, and how much historical validation the current methodology carries. A thin-data score is labeled as thin rather than smoothed over.

**Data sufficiency** reports whether enough of the weighted rubric has been verified to trust the number. Only protocols that clear the sufficient threshold are published at all. A protocol absent from the library is one we will not yet put a number on, which is a permanent rule rather than a temporary gap.

**Bands** are relative to the currently covered set, not absolute grades. Elevated means a score sits above the statistical norm of the protocols scored so far. Because the comparison set grows as coverage grows, band boundaries can move over time, and a protocol's band can change without its own score changing.

## How the rubric is tested

Before anything ships, the rubric is scored against real history point-in-time: each protocol is scored using only what was knowable the day before an incident, then compared against protocols that were never hit in the same window. Under that test, protocols with a prior incident reached a second one at roughly 27 times the rate of protocols with no incident history at all. The full backtest report, including its coverage limits, is published separately.

## What a score does not do

**It does not predict.** A score does not predict whether or when a protocol will be exploited. It measures the presence and severity of risk signals, which is a different claim.

**It does not measure market risk.** Price movement, depegging, yield sustainability, and liquidity conditions are outside the rubric entirely.

**It does not assess anything off-chain.** Custodial arrangements, centralized venues, and legal or regulatory exposure are not scored.

**It is not advice.** A score is not a substitute for your own due diligence, and it is not financial, legal, or insurance advice.

## Versioning

Every score is tagged with the exact methodology version that produced it. Any change to how a category is weighted or computed gets its own documented, dated version bump before it is ever run against real data, so a score is always traceable to the specific rubric behind it, and nothing changes retroactively under a score already published.

---

*The full internal rubric - exact category weights, the scoring rules that convert each fact into a number, threshold values, and rule-by-rule justifications - is Foreshock's working methodology document and is not published. What is above is an accurate description of how scores are built, not a marketing simplification of it.*