The following post outlines a LEGO initiative to allocate treasury funds towards an Immunefi bug bounty program.

Through LEGO, bug bounty within the allotted budget for submissions of Boulder size and below will be honoured, whilst bug bounties outside of budget (or of mountain level) will initiate a vote to pass. The LEGO classification system can be found in more detail here.

Immunefi is a proven bug bounty protocol for smart contract based applications through which security researchers, developers and more can review code and disclose vulnerabilities for compensation.

The bug bounty program covers its smart contracts and apps and is focused on the prevention of loss of user funds, denial of service, governance hijacks, data breaches, and data leaks.

The precise details of the bug bounty are disclosed below.

Rewards by Threat Level

Rewards are distributed according to the impact of the vulnerability based on theImmunefi Vulnerability Severity Classification System (IVSCS).

This is a simplified 5-level scale, with separate scales for websites/apps and smart contracts/blockchains, encompassing everything from consequence of exploitation to privilege required to likelihood of a successful exploit.

Smart Contracts and Blockchain

- Critical USD 100,000
- High USD 20,000
- Medium USD 5,000
- Low USD 1,000

Website and Apps

- Critical USD 20,000
- High USD 7,500
- Medium USD 3,250
- Low USD 500

All web and app bugs must come with a PoC in order to be accepted. All web and app bug reports without a PoC will be rejected with a request for a PoC.

Payouts are handled by the LEGO team directly and are denominated in USD. Payouts can be done in ETH, DAI, RAI, or LDO, at the decision of the bug bounty hunter.

Assets in Scope

Target

Type

Deployed Contracts | Lido Docs

Smart Contract

GitHub - lidofinance/dc4bc: distributed custody for the beacon chain

Smart Contract

GitHub - lidofinance/steth-price-feed: Price feed for stETH/ETH pair

Smart Contract

GitHub - lidofinance/curve-merkle-oracle: Trustless oracle for a Curve ETH/stETH pool using MPT proofs

Smart Contract

GitHub - lidofinance/lido-dao: Lido DAO smart contracts

Smart Contract

https://lido.fi

Website/App

https://stake.lido.fi

Website/App

Smart Contracts labeled or categorized as testnet are not in scope of this bug bounty program.

Subdomains other than the one listed in this table are not included in the bug bounty program scope.

Prioritized vulnerabilities

We are especially interested in receiving and rewarding vulnerabilities of the following types:

Smart Contracts and Blockchain

- Re-entrancy
- Logic errors
- including user authentication errors
- including user authentication errors
- Solidity/EVM details not considered
- including integer over-/under-flow
- Including rounding errors
- including unhandled exceptions
- including integer over-/under-flow
- Including rounding errors
- including unhandled exceptions
- Trusting trust/dependency vulnerabilities
- including composability vulnerabilities
- including composability vulnerabilities
- Oracle failure/manipulation
- Novel governance attacks
- Economic/financial attacks
- including flash loan attacks
- including flash loan attacks
- Congestion and scalability
- including running out of gas
- including block stuffing
- including susceptibility to frontrunning
- including running out of gas
- including block stuffing
- including susceptibility to frontrunning
- Consensus failures
- Cryptography problems

- Signature malleability
- Susceptibility to replay attacks
- Weak randomness
- Weak encryption
- Signature malleability
- Susceptibility to replay attacks
- Weak randomness
- Weak encryption
- Susceptibility to block timestamp manipulation
- Missing access controls / unprotected internal or debugging interfaces

Websites and Apps

- Remote Code Execution
- Trusting trust/dependency vulnerabilities
- Vertical Privilege Escalation
- XML External Entities Injection
- SQL Injection
- LFI/RFI
- Horizontal Privilege Escalation
- Stored XSS
- Reflective XSS with impact
- CSRF with impact
- Direct object reference
- Internal SSRF
- Session fixation
- Insecure Deserialization
- DOM XSS
- SSL misconfigurations
- SSL/TLS issues (weak crypto, improper setup)
- URL redirect
- Clickjacking (must be accompanied with PoC)
- Misleading Unicode text (e.g. using right to left override characters)

Out of Scope & Rules

The following vulnerabilities are excluded from the rewards for this bug bounty program:

- Attacks that the reporter has already exploited themselves, leading to damage
- Attacks requiring access to leaked keys/credentials
- Attacks requiring access to privileged addresses (governance, strategist)

Smart Contracts and Blockchain

- Incorrect data supplied by third party oracles
- Not to exclude oracle manipulation/flash loan attacks
- Not to exclude oracle manipulation/flash loan attacks
- Basic economic governance attacks (e.g. 51% attack)
- Lack of liquidity
- Best practice critiques
- Sybil attacks

Websites and Apps

- Theoretical vulnerabilities without any proof or demonstration
- Content spoofing / Text injection issues
- Self-XSS
- Captcha bypass using OCR
- CSRF with no security impact (logout CSRF, change language, etc.)
- Missing HTTP Security Headers (such as X-FRAME-OPTIONS) or cookie security flags (such as “httponly”)
- Server-side information disclosure such as IPs, server names, and most stack traces
- Vulnerabilities used to enumerate or confirm the existence of users or tenants
- Vulnerabilities requiring unlikely user actions
- URL Redirects (unless combined with another vulnerability to produce a more severe vulnerability)
- Lack of SSL/TLS best practices
- DDoS vulnerabilities
- Attacks requiring privileged access from within the organization
- Feature requests
- Best practices

The following activities are prohibited by bug bounty program:

- Any testing with mainnet or public testnet contracts; all testing should be done on private testnets
- Any testing with pricing oracles or third party smart contracts
- Attempting phishing or other social engineering attacks
- Any testing with third party systems and applications (e.g. browser extensions) as well as websites (e.g. SSO providers, advertising networks)
- Any denial of service attacks
- Automated testing of services that generates significant amounts of traffic
- Public disclosure of an unpatched vulnerability in an embargoed bounty