Audit Competition

Summary Report
November 2024

Content Index

Overview

Immunefi Audit Competitions are special, time-limited events that supercharge the reach and visibility of programs to our whitehat community.

From November 13 to December 6, 2024, the Celo Competition offered up to $50,000 USD in rewards for security researchers.

Immunefi's Discord server hosted a channel for enhanced, two-way communication between whitehats and the Celo team, improving feedback and response times. Managed Triaging was also activated for the duration of this event, streamlining the resolution process for incoming bug reports.

During this event, 1 low/medium bugs, 3 high, 3 critical, and 1 insight reports were found on the target contracts. A total of 22 security researchers participated.

Celo distributed the $50k reward pool to 8 of the very best submissions for the security researchers' valiant efforts, including "Insight" submissions scored on a rating system that takes into account levels of:

1) Security best practices
2) Code optimizations and enhancements
3) Architectural decentralization and composability
4) Documentation improvements

Celo Introduction

Celo is on a mission is to build a regenerative digital economy that creates conditions of prosperity for all.

For more information about Celo, please visit https://celo.org/

Celo provides rewards in cUSD, denominated in USD.

Scope Of Assets

The target assets in scope for the Audit Competition included Celo's smart contracts, specifically the Staked contracts and the Multisig architecture.

The total nSLOC was 5,253.

Summary

Duration: **Three weeks**	Comp Date: **13 Nov - 6 Dec 2024**	Rewards Pool: **$50,000**	nSLOC: **5,253**
Submitted reports: **57**	Security researchers: **22**	Valid vulnerabilities: **7**	Insight reports: **1**

Total Whitehat Participation

22
Total Researchers

4
Paid Researchers

Leaderboard

Position	Reward	Username	Valids	Insights
1	$20,767	innertia	3	0
2	$15,387	jovi	2	1
3	$10,309	shadowHunter	1	0
4	$3,535	okmxuse	1	0

Top 3 Reports

Fraudulent padding of governance voting power

Report number: 37251

Submitted by: @innertia

Target: https://github.com/celo-org/celo-monorepo/blob/release/core-contracts/12/packages/protocol/contracts/governance/LockedGold.sol

Impacts:

- Manipulation of governance voting results deviating from the outcome and resulting in direct change from intended effect of original results

Program Action: Confirmed as critical severity.

Report Excerpt:

A `Slash` reduces or increases a user's nonvoting balance. However, there is no processing related to `delegate`. This can cause various vulnerabilities including; Inflating the number of votes and Withdrawing tokens while maintaining the number of votes etc.

Rollback of the incorrect state interferes with the progress of the epoch process, prevents the user from receiving rewards, blocks the launch of the associated contract function etc

Report number: 37010

Submitted by: @innertia

Target:
https://github.com/celo-org/celo-monorepo/blob/release/core-contracts/12/packages/protocol/contracts-0.8/common/EpochManager.sol

Impacts:
- Temporary freezing of funds
- Smart contract unable to operate due to lack of token funds
- Griefing - Issuance of an unauthorized amount of StableToken (not to the attacker)
- Blocking of execution of functions of other contracts

Program Action: Confirmed as high severity.

Report Excerpt:

`EpochProcess` is managed by `EpochProcessStatus`, which has three statuses: `NotStarted`, `Started`, and `IndivudualGroupsProcessing`. They should change and cycle in this order, and each status has a different function that can be started.

However, after being changed from `Started` to `IndivudualGroupsProcessing`, there is a function launch route that reverts back to `Started`. This causes various problems such as prevents the user from receiving rewards, blocks the launch of the associated contract function, etc.

Overflow due to lack of checks leading to incorrect price calculations

Report number: 37206

Submitted by: @okmxuse

Target:
https://github.com/celo-org/optimism/blob/celo10/op-chain-ops/cmd/check-derivation/main.go

Impacts:

- Smart contract unable to operate due to lack of token funds

Program Action: Confirmed as medium severity.

Report Excerpt:

Inside the `check-derivation/main.go` function, the `getRandomSignedTransaction` function is invoked (note that this function traces all the way back to `checkConsolidation` which is then called in the `main.go` function).
`getRandomSignedTransaction` calls `IntrinsicGas` at three places. We will focus on the one that includes the `accessList`, which is `case types.AccessListTxType`

End of Report