Ethena Security Review

Pashov Audit Group

Conducted by: Dan Ogurtsov, immeas

December 19th 2023 - December 22nd 2023

Contents

1. About Pashov Audit Group

Pashov Audit Group consists of multiple teams of some of the best smart contract security researchers in the space. Having a combined reported security vulnerabilities count of over 1000, the group strives to create the absolute very best audit journey possible - although 100% security can never be guaranteed, we do guarantee the best efforts of our experienced researchers for your blockchain protocol. Check our previous work here or reach out on Twitter @pashovkrum.

2. Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

3. Introduction

A time-boxed security review of the **ethena** repository was done by **Pashov Audit Group**, with a focus on the security aspects of the application's smart contracts implementation.

4. About ethena

Copied from the first security review

The Ethena protocol is building `USDe` which will be a synthetic dollar with yield bearing properties, deployed on Ethereum. The stablecoin will be 100% collateralized with no collateral within the banking system, using as collateral USDC, stETH and other LSDs. The yield is expected to come from `stETH` and arbitrage. The `USDe` smart contract's minting and redeeming will be handled in a trusted manner by the Ethena team.

More docs

Continued

The protocol has now added the `ENA` governance token as well as LP staking functionality to incentivize people to provide liquidity into the Ethena pools.

5. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

5.1. Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

5.2. Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5.3. Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

6. Security Assessment Summary

***review commit hash* - d74901ab42048e142ad53cf4cdfa98a5a82c4ef7**

Scope

The following smart contracts were in scope of the audit:

- `EthenaLPStaking`
- `ENA`

7. Executive Summary

Over the course of the security review, Dan Ogurtsov, immeas engaged with Ethena to review ethena. In this period of time a total of **1** issues were uncovered.

Protocol Summary

Protocol Name	ethena
Repository	https://github.com/ethena-labs/ethena/
Date	December 19th 2023 - December 22nd 2023
Protocol Type	synthetic dollar

Findings Count

Severity	**Amount**
Low	1
Total Findings	**1**

Summary of Findings

ID	**Title**	**Severity**	**Status**
[L-01]	Multiple cooldowns are not managed for the same user	Low	Acknowledged

8. Findings

8.1. Low Findings

[L-01] Multiple cooldowns are not managed for the same user

When `unstake()` a given user updates their `cooldownStartTimestamp`.

```
...
StakeData storage stakeData = stakes[msg.sender][token];
...
stakeData.cooldownStartTimestamp = uint104(block.timestamp);
```

It means different unstakes by the user will not go in parallel - only the last unstake will accumulate all previous pending unstakes and can be withdrawn on `last unstake time + stakeParameters.cooldown`. It is also relevant for those unstakes that waited for enough `cooldown` and are ready to withdraw - such unstakes will wait for a new `cooldown` if some new unstake is called.

If this behavior is not desired, consider managing a separate queue for unstakes and withdrawals, where every unstake has its own storage.