HoneyFun Security Review

March 25, 2025 - March 26, 2025

Date: March 26, 2025

Conducted by: **KeySecurity**
gkrastenov, Lead Security Researcher

Table of Contents

1 About KeySecurity

KeySecurity is a innovative Web3 security company that hires top-talented security researchers for your project. We have conducted over 40+ security reviews for various projects, collectively holding over $300,000,000 in TVL. For security audit inquiries, you can reach out to us on Twitter/X or Telegram `@gkrastenov` or check our previous work `here`.

2 About HoneyFun

`Honeyfun AI` is pioneering the co-ownership framework for AI agents specifically tailored for the Berachain ecosystem, focusing on defi, gaming and entertainment. We envision AI agents as pivotal revenue-generating entities in the future, as we believe the era of Utility AI Agents is just beginning, and in the coming years, their untapped potential across every field will be revealed.

3 Disclaimer

Audits are a time, resource, and expertise bound effort where trained experts evaluate smart contracts using a combination of automated and manual techniques to identify as many vulnerabilities as possible. Audits can show the presence of vulnerabilities **but not their absence**.

4 Risk classification

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

4.1 Impact

- **High** - leads to a significant loss of assets in the protocol or significantly harms a group of users.
- **Medium** - only a small amount of funds can be lost or a functionality of the protocol is affected.
- **Low** - any kind of unexpected behaviour that's not so critical.

4.2 Likelihood

- **High** - direct attack vector; the cost is relatively low to the amount of funds that can be lost.
- **Medium** - only conditionally incentivized attack vector, but still relatively likely.
- **Low** - too many or too unlikely assumptions; provides little or no incentive.

4.3 Actions required by severity level

- **Critical** - client **must** fix the issue.
- **High** - client **must** fix the issue.
- **Medium** - client **should** fix the issue.
- **Low** - client **could** fix the issue.

5 Executive summary

Overview

Project Name	HoneyFun Meme Contract
Repository	https://github.com/honey-fun/honey-fun-contracts
Commit hash	7274b4fc1da2120578b9aa14b6efc60ec6eef227
Review Commit hash	56dace7bd175c3eb9fe0c96ec5b9f84199aa066c
Documentation	https://docs.honey.fun/
Methods	Manual review

Scope

Meme.sol

Timeline

March 25, 2025	Audit kick-off
March 25, 2025	Preliminary report
March 26, 2025	Mitigation review

Issues Found

Severity	**Count**
High	0
Medium	0
Low	0
Information	2
Total	**2**

6 Findings

6.1 Information

6.1.1 Emit event in crucial place

Severity: *Information*

Context: Meme.sol#L69

Description: Emit an event in crucial places, such as in the `makeTransferable()` function, when the transferable functionality of the meme is set to true.

```
    function makeTransferable() external override onlyTransferControllerAccess {
      isTransferable = true;
      //@audit-issue emit event
    }
```

Recommendation: Emit event in `makeTransferable()` function.

Resolution and Client comment: Resolved.

6.1.2 Use _totalSupply directly instead of calling totalSupply() internally

Severity: *Information*

Context: Meme.sol#L63

Description: In the constructor of the `Meme` contract, the total supply of the token is provided directly. Use the `_totalSupply` variable instead of internally calling the `totalSupply()` function when the initial supply is minted.

```
_mint(_mintTo, totalSupply()); //@audit-issue use directly _totalSupply
```

Recommendation: Use `_totalSupply` directly instead of calling `totalSupply()` internally.

Resolution and Client comment: Resolved.