shieldify

Honeypot Finance
Pot2Pump Updates

SECURITY REVIEW

Date: 3 February 2025

CONTENTS

1. About Shieldify

Positioned as the first hybrid Web3 Security company, Shieldify shakes things up with a unique subscription-based auditing model that entitles the customer to unlimited audits within its duration, as well as top-notch service quality thanks to a disruptive 6-layered security approach. The company works with very well-established researchers in the space and has secured multiple millions in TVL across protocols, also can audit codebases written in Solidity, Rust, Go, Vyper, Move and Cairo.

Learn more about us at shieldify.org.

2. Disclaimer

This security review does not guarantee bulletproof protection against a hack or exploit. Smart contracts are a novel technological feat with many known and unknown risks. The protocol, which this report is intended for, indemnifies Shieldify Security against any responsibility for any misbehavior, bugs, or exploits affecting the audited code during any part of the project's life cycle. It is also pivotal to acknowledge that modifications made to the audited code, including fixes for the issues described in this report, may introduce new problems and necessitate additional auditing.

3. About Honeypot Finance – Pot2Pump Updates

Honeypot Finance acts as a Proof-of-Liquidity (PoL) Accelerator that unites a fair launchpad, Dreampad, and a secure DEX, Henlo DEX:

- Dreampad supports our innovative Fair Token Offering (FTO) model and both Fjord Foundry's LBP and Fixed Price Sales to ensure successful and sustainable token launches for projects.
- Henlo DEX is powered by the A2MM protocol, enables automatic liquidity deployment and maximizes liquidity utilization for traders and investors.
- Pot2Pump combines all the advantages of the FTO model, with specific adjustments for meme tokens and protection against bots. Their PoL Accelerator aims to embody the aspirations of the Berachain community by providing a comprehensive suite of DeFi tools. These tools are crafted to empower individuals with financial autonomy. Their unique flywheel operates on a community-driven paradigm, fostering an ecosystem of protocols and validators where increased engagement leads to enhanced liquidity.

Learn more about Honeypot(Pot2Pump)'s concept and the technicalities behind it here.

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** - results in a significant risk for the protocol's overall well-being. Affects all or most users

- **Medium** – results in a non-critical risk for the protocol affects all or only a subset of users, but is still unacceptable
- **Low** – losses will be limited but bearable – and covers vectors similar to griefing attacks that can be easily repaired

4.2 Likelihood

- **High** – almost certain to happen and highly lucrative for execution by malicious actors
- **Medium** – still relatively likely, although only conditionally possible
- **Low** – requires a unique set of circumstances and poses non-lucrative cost-of-execution to rewards ratio for the actor

5. Security Review Summary

The security review lasted 2 days with a total of 32 hours dedicated by 2 researchers from the Shield-ify team.

Overall, the code is well-written. The audit report contributed by identifying one Low-severity issue for missing call `_disableInitializers()` function in the constructor.

The Honeypot team has done an excellent job on the development, demonstrating both expertise and dedication.

5.1 Protocol Summary

Project Name	Honeypot Finance – Pot2Pump Updates
Repository	HoneyPot-MemeLaunchPad-Algebra
Type of Project	DeFi, Proof-of-Liquidity (PoL) Accelerator
Audit Timeline	2 days
Review Commit Hash	e51a68bfbdc591b643281b82d0e87a42ec2024ff
Fixes Review Commit Hash	428c4b96378943fb2a0d1ad5a380d96d280e1300

5.2 Scope

The following smart contracts were in the scope of the security review:

File	nSLOC
contracts/hpot/core/pot2pump/Pot2PumpFactory.sol	130
contracts/hpot/core/pot2pump/Pot2PumpPairV2.sol	74
contracts/hpot/core/pot2pump/Pot2PumpPairV1.sol	65
contracts/hpot/core/pot2pump/interfaces/IPot2PumpFactory.sol	16

contracts/hpot/core/pot2pump/interfaces/IPot2PumpPair.sol	7
contracts/hpot/core/berascout/BeraScoutFactory.sol	197
contracts/hpot/core/berascout/BeraScoutPair.sol	155
contracts/hpot/core/berascout/interfaces/IBeraScoutFactory.sol	31
contracts/hpot/core/berascout/interfaces/IBeraScoutPair.sol	6
Total	**681**

6. Findings Summary

The following number of issues have been identified, sorted by their severity:

- **Low** issues: **1**

ID	Title	Severity	Status
[L-01]	Call DisableInitializer In The Constructor To Prevent Anyone From Initializing The Implementation Contract	Low	Fixed

7. Findings

[L-01] Call DisableInitializer In The Constructor To Prevent Anyone From Initializing The Implementation Contract

Severity

Low Risk

Description

When using UUPSUpgradeable, best practice is to call `_disableInitizliers()` in the constructor since the initializer function in the implementation contract can be called by anyone.

Recommendation

Set `_disableInitializers()` in the constructor.
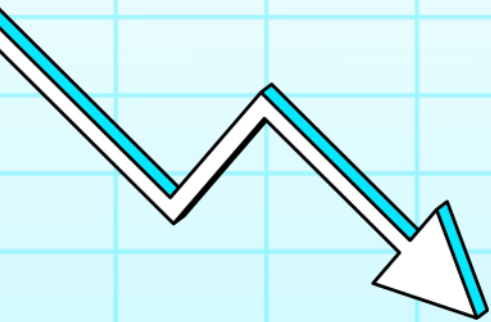
```
constructor() {
  _disableInitializers();
}
```

Team Response

Fixed.

shieldify

Thank you!