ETHOS AUDIT

Aping DRIP

Audit Report

May 3, 2022

Contents

Executive Summary

Audit Details

Project Name	ApingDRIP
Codebase	https://www.bscscan.com/address/0x0d0918B9c41c2eF82E96E1908b776B75c867851e#code
Source Code	ApringDRIP.sol
Initial Audit Date	May 1, 2022
Revision Dates	-
Methodology	Manual

Methodology

This audit's objectives are to evaluate:

- Security-related issues
- Code quality
- Relevant documentation
- Adherence to specifications
- Adherence to best practices

This audit examines the possibility of issues existing along the following vectors (but not limited to):

- Single & Cross-Function Reentrancy
- Front Running (Transaction Order Dependence)
- Timestamp dependence
- Integer Overflow and Underflow
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Number rounding errors
- DoS with (Unexpected) Revert
- DoS with Block Gas Limit

- Insufficient gas griefing
- Forcibly sending native currency
- Logical oversights
- Access control
- Centralization of power
- Logic-Specification Contradiction
- Functionality duplication
- Malicious token minting

The code review conducted for this audit follows the following structure:

1. Review of specifications, documentation to assess smart contract functionality
2. Manual, line-by-line review of code
3. Code's adherence to functionality as presented by documentation
4. Automated tool-driven review of smart contract functionality
5. Assess adherence to best practices
6. Provide actionable recommendations

Contract Details

Contract ID	0x0d0918B9c41c2eF82E96E1908b776B75c867851e
Network	Binance Smart Chain
Language	Solidity
Compiler	v0.8.9+commit.e5eed63a
Verification Date	May 2, 2022
Contract Type	Utility Contract
Libraries	Custom

Result Summary

Ethos' audit of the **ApingDRIP** smart contract has concluded with a **POSITIVE** result. The initial review identified a number of non-critical issues and one infromational issue. The remaining report includes all issues identified in the initial review, as well as the revised status post resolution by the team.

- The smart contract is a variant of the 'miner' meta

- It allows users to deposit **DRIP-BUSD Pancakeswap LP tokens** into the contract

- Deposits are locked on deposit and redistributed to contract participants over time

- The rate of redistributions approximately **6% daily** and varies based on the rate of increase of total value locked

- There is a referral bonus distributed to referrers of approximately **6.25%**

- There is an **6% dev fee** applied on all deposits and withdrawls

- Value locked within the contract **cannot be manually removed** by the owner

- The **contract cannot be closed or shut off** at any point after deployment

To conclude, this smart contract functions as it is designed. It is not ruggable by the owner or any other entities through attack vectors currently known in the EVM community.

Issues Reported

Severity	Unresolved	Acknowledged	Resolved
Extreme	0	0	0
High	0	0	0
Medium	0	0	0
Low	0	4	0

Issues Summary

ID	Title	Severity	Status
AA-0	Variables can be declared as 'constant'	Low	Acknowledged
AA-1	Missing Event emissions	Low	Acknowledged
AA-2	Function initializing state	Low	Acknowledged
AA-3	Functions that could be declared external	Low	Acknowledged

Detailed Findings

AD-0 – Variables can be declared as 'constant'

Severity: Low   **Status**: Acknowledged

Description: Several global variables that are initialized with a value on definition can be declared as constant since they are never changed throughout the body of the smart contract.

Risk: This is a minor gas optimization issue.

Recommendation: We recommend declaring these variables as 'constant' if they aren't going to be changed.

AD-1 – Missing Event emissions

Severity: Low   **Status**: Acknowledged

Description: There are several functions that change state variables, however, they do not emit events to pass the changes out of chain.

Risk: Not emitting an event from functions that impose changes to state variables could result in a lack of functionality often required for sound logic and functionality within external applications calling on smart contract functions.

Recommendation: We recommend emitting events for all essential state variables that are possible to be changed during runtime.

AD-2 – Function initializing state

Severity: Low   **Status**: Acknowledged

Description: A state variable is initialized through function calls that are not pure/constant, or that use non-constant state variables

Risk: Users might intend a function to return a value a state variable can initialize with, without realizing the context for the contract is not fully initialized.

Recommendation: Remove any initialization of state variables via non-constant state variables or function calls. If variables must be set upon contract deployment, locate initialization in the constructor instead.

AD-3 – Functions that could be declared external

Severity: Low   **Status**: Acknowledged

Description: Several functions are declared as public visibility, however, since they are never called by the contract they should be declared external.

Risk: This is a gas optimization issue.

Recommendation: We recommend that functions that are never called by the contract to be declared as external to save gas.

Code Documentation

The code has a minimal amount of comments. This could be improved in order to help others understand the contract.

Adherence to Specifications

The smart contract does adhere to the smart contract functionality described by the ApingDRIP team.