ETHOS
AUDIT

Panda Farms

Audit Report

Apr. 20, 2022

Contents

Executive Summary 3
Audit Details 3
Methodology 3
Contract Details 4
Result Summary 4
Issues Reported 5
Issues Summary 5
Detailed Findings 5
PF-0 – Variables can be declared as 'constant' 5
PF-1 – Missing Event emissions 6
PF-2 – Function initializing state 6
PF-3 – Functions that could be declared external 6
Code Documentation 7
Adherence to Specifications 7
Adherence to Best Practices 7
Appendix 7
Functions 7
Global Variables 8

Executive Summary

Audit Details

Project Name	Panda Farms
Codebase	https://www.bscscan.com/address/0x5f74970cc63f002ddf511891b6955a58d48b65a4#code
Source Code	PandaFarms.sol
Initial Audit Date	April 20, 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	0x5F74970cc63F002DdF511891B6955a58D48B65A4
Network	BSC
Language	Solidity
Compiler	v0.8.9+commit.e5eed63a
Verification Date	Apr. 20, 2022
Contract Type	Utility Contract
Libraries	OpenZeppelin

Result Summary

Ethos' audit of the Panda Farms smart contract has concluded with a **POSITIVE** result. The initial review identified a number of non-critical issues. 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 network native tokens into the contract
- Deposits are locked on deposit and redistributed to users over time
- The rate of redistributions approximately **8% daily** and varies based on the rate of increase of total value locked
- There is a referral bonus distributed to referrers of approximately **12.5%**
- There is a **5% dev fee** applied on all deposits and redistributions
- 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 does what it is designed to, and 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
PF-0	Variables can be declared as ‘constant’	Low	Acknowledged
PF -1	Missing Event emissions	Low	Acknowledged
PF -2	Function initializing state	Low	Acknowledged
PF -3	Functions that could be declared external	Low	Acknowledged

Detailed Findings

PF-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.

PF-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.

PF-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.

PF-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 adheres to the smart contract functionality described by the Panda Farms team and is in line with its intended usage.

Adherence to Best Practices

The smart contract adheres to the majority of best practices associated with a standard BEP-20 token. The few things that don't follow the best practices are noted in the automated review.

Appendix

Functions

Function	Parameters	Visibility	Modifiers	Returns	Requires	Events
constructor						
hatchEggs	address ref	public			initialized	
sellEggs		public			initialized	
beanRewards	address ref	public		uint256		
buyEggs	address ref	public	payable		initialized	
calculateTrade	uint256 rt, uint256 rs, uint256 bs	private		uint256		
calculateEggSell	uint256 eggs	public		uint256		
calculateEggBuy	uint256 eth,uint256 contractBalance	public		uint256		
calculateEggBuySimple	uint256 eth	public		uint256		
devFee	uint256 amount	private		uint256		
seedMarket		public	payable onlyOwner		marketEggs == 0	
getBalance		public		uint256		
getMyMiners	address adr	public		uint256		
getMyEggs	address adr	public		uint256		
getEggsSinceLastHatch	address adr	public		uint256		
min	uint256 a, uint256 b	private		uint256		

Global Variables

Variable	Type	Visibilty	Read by Functions	Written by Functions
EGGS_TO_HATCH_1MINERS	uint256	private		
PSN	uint256	private		
PSNH	uint256	private		
devFeeVal	uint256	private		
initialized	bool	private		
recAdd	address payable	private		
hatcheryMiners	mapping (address => uint256)	private		
claimedEggs	mapping (address => uint256)	private		
lastHatch	mapping (address => uint256)	private		
referrals	mapping (address => address)	private		
marketEggs	uint256	private		