MAY 19th, 2023

ZKBOOST

Audit Report

Ensuring Code Integrity – Our Audit Seal of Trust

Bernie Duck

HTTPS://T.ME/ZKBOOST

@ZKBOOST_FINANCE

DISCLAIMER

1. This audit report is based on the information provided to us and our assessment of the code at the time of the audit. Changes made after the audit may impact the security or functionality of the contract.

2. The audit report does not constitute financial, legal, or investment advice. It is solely for informational purposes and should not be relied upon for making any decisions or investments.

3. While we strive to conduct a thorough and comprehensive audit, it is not possible to guarantee the absence of all vulnerabilities or risks in the audited contract.

4. The audit report is limited to the code and contract under review. It does not cover the project's team, business model, or future performance.

5. Our audit findings and recommendations are based on our expertise and understanding of industry best practices at the time of the audit. However, security risks and technologies evolve, and future developments may impact the validity of our findings.

6. We do not take any responsibility for any losses or damages incurred as a result of using the audited contract or relying on the information provided in the audit report.

7. The audit report may contain technical terms and concepts. It is recommended to seek professional advice if any clarification or further explanation is needed.

8. Any third-party tools, libraries, or services used in the audited contract are not audited unless explicitly stated in the report.

9. The audit report is confidential and intended solely for the named recipient. It should not be shared or distributed without prior written consent from us.

10. We reserve the right to update or amend the audit report if new information or significant changes to the contract occur.

Please note that the above disclaimers are for reference only, and it is recommended to consult with legal professionals to ensure compliance with applicable laws and regulations in your jurisdiction.

Network

https://zksync.io/

Website

https://bernieduck.com/

Twitter

https://twitter.com/Bernie_Duck

Medium

Description

Bernie Duck is the quacktastic cryptocurrency that combines the charm of Donald Duck with the power of memes. Built on zkSync ERA, Bernie Duck offers lightning-fast transactions and a seamless user experience. Join our vibrant community and embark on a hilarious journey towards financial freedom. Get ready to quack your way to the moon with Bernie Duck

Project Engagement

During the Date of 24 May 2023, Bernie Duck Team engaged zkBoost to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided zkBoost with access to their code repository and whitepaper.

Logo

Contract Link
v1.0

https://explorer.zksync.io/address/0x9290A318D83CD12B47a452CD590FE7213E0073EC

Note for Investors: We only Audited $BERNIE token contract. However, If the project has other contracts (for example, a Presale contract etc),and they were not provided to us in the audit scope then we cannot comment on its security and we are not responsible for it in any way.

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impactof that eventon the organization or system.RiskLevel is computedbased on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 – 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon aspossible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:
1. Code review that includes the following:
i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.

ii) Manual review of code, which is the process of reading source code line-by- line in an attempt to identify potential vulnerabilities.

iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to zkBoost describe.

2. Testing and automated analysis that includes the following:
i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.

ii) Symbolic execution, which is analyzing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Scope of Work/Verify Claims

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Is contract an upgradeable
2. Correct implementation of Token standard
3. Deployer cannot mint any new tokens
4. Deployer cannot burn or lock user funds
5. Deployer cannot pausethe contract
6. Deployer cannot set fees
7. Deployer cannot blacklist/antisnipe addresses
8. Overall checkup (Smart Contract Security)

Write functions of contract

v1.0

Write

1. approve
2. claimStuckTokens
3. decreaseAllowance
4. increaseAllowance
5. renounceOwnership
6. setAMMPair
7. setWhiteList
8. transfer
9. transferFrom
10. transferOwnership

Is contract an Upgradeable

v1.0

Name	
Is contract an Upgradeable ?	NO

Correct implementation of Token standard

ERC-20		
Function	**Description**	**Verify**
TotalSupply	Provides information about the total token supply	OK
BalanceOf	Provides account balance of the owner's account	OK
Transfer	Executes transfers of a specified number of tokens to a specified address	OK
TransferFrom	Executes transfers of a specified number of tokens from a specified address	OK
Approve	Allow a spender to withdraw a set number of tokens from a specified account	OK
Allowance	Returns a set number of tokens from a spender to the owner	OK

Deployer cannot mint any new tokens

Name	
Deployer cannot mint any new tokens	✓

Deployer cannot burn or lock user funds

Name	
Deployer cannot burn or lock user funds	✓

Deployer cannot pause the contract

Name	
Deployer cannot pause the contract	✓

Deployer cannot set fees

Name	
Deployer cannot set fees	✓

Deployer cannot blacklist/anti-snipe addressed

Name	
Deployer cannot blacklist/anti-snipe addressed	✓

Overall checkup

Name	
User cannot transfer token until the owner of contract use the function "setAMMPair" with non-zero Address argument	✓

```
if (AMMpair == address(0)) {
    require(isWhiteList(from) || isWhiteList(to), "not allowed yet");
}
```

Audit Results

Critical issues

No Critical issues

Critical issues

No Critical issues

Resolved Critical issues

Note:

Medium issues

No Medium Issues

Low issues

No Low issues

Information

No Information issues

Final result

PASS

MAY 19th, 2023

ZKBOOST

ZK Boost

HTTPS://T.ME/ZKBOOST

@ZKBOOST_FINANCE