SPHINXSHIELD

SECURITY AUDIT REPORT

Audit Request
19 June 2023

Audit Onboard
19 June 2023

Revision At
No Revision

Release At
21 June 2023

OVERVIEW

This audit report has been prepared by **SphinxShield** for **Obama Inu**. The purpose of this audit is to provide a comprehensive review of the project's source code security. The evaluation aims to identify potential vulnerabilities and risks, ensuring the project's solidity codebase adheres to best practices and industry standards.

SphinxShield's objective is to assess the security of the project's smart contract, mitigating potential security breaches and protecting investors' interests. The audit covers critical aspects, including the thorough analysis of the contract's source code, ownership structure, tokenomics, and compliance with security measures. Additionally, the assessment extends to external dependencies and potential risks associated with the smart contract implementation.

This audit report serves as a valuable resource for investors, enabling them to make well-informed decisions based on the project's security posture. However, it is essential to note that while **SphinxShield's** evaluation provides insights into the project's source code security, it does not guarantee or reflect the project's ultimate outcome and goals.

"The results of this audit are purely based
on the team's evaluation and does not
guarantee nor reflect the projects
outcome and goal."

– SphinxShield's Team –

TABLE OF CONTENTS

Obama Inu

PROJECT DESCRIPTION

According to their whitepaper or website:
Obama Inu is a BSC token inspired by President Barack Obama's principles of progress and social justice. It is a community-driven project focused on decentralized finance and aims to empower individuals through blockchain technology. Obama Inu supports short & long term growth initiatives and encourages community governance, giving token holders a voice in the project's direction. Obama Inu promotes fair distribution, transparency, and security. By joining Obama Inu, you can be apart of a movement that strives for positive change and embodies the values of equality and empowerment envisioned by President Obama.

Presale started on **15 June 2023**
DEX listed on **17 June 2023**
Category: **Meme Token**

CONTRACT INFORMATIONS

INDICATIOR	VALUE
Token Name	Obama Inu
Token Symbol	OBAMA
Total Supply	100,000,000
Already Listed On Dex	Already Listed
Dex Listed	PancakeV2
Open Source	1
Price	0.00005499 (-0.093%)
Volume 24H	$82.91
Liquidity	$1,247 (4 BNB)
Txn Count	9 (-0.31%)
Market Cap	$5,498 (0.000%)

SECURITY INFORMATIONS

INDICATIOR	VALUE
Honeypot	Not Detected
Buy Fees	5.00%
Sell Fees	4.99%
Holder Count	80

SphinxShield's Contract Review Process

At SphinxShield, our contract review process follows a meticulous approach, leveraging a combination of industry-leading tools and expert analysis. We pay special attention to the following key areas:

Comprehensive Vulnerability Testing:

- We employ both automated and manual testing techniques to identify common and uncommon vulnerabilities in the smart contracts.
- Tools such as Hardhat, Slither, and OpenZeppelin are utilized to detect potential security flaws and ensure the contracts are resilient against attacks.

Compliance with Best Practices and Standards:

- Our team assesses the codebase to ensure it adheres to current best practices and industry standards for solidity development.
- We utilize tools like Solidity Compile to verify that the contract code follows recommended coding conventions and style guidelines.

Alignment with Client's Specifications:

- We carefully review the contract logic to ensure it aligns with the specifications and intentions of the client.
- Our experts analyze the contract's functionality and verify that it meets the desired requirements without any unintended behaviors.

Industry Benchmarking:

- We conduct a thorough cross-referencing analysis, comparing the contract structure and implementation against similar smart contracts developed by industry leaders.
- This benchmarking helps us identify any deviations or potential issues that may arise from differences in contract design or implementation choices.

Manual Codebase Review:

- Our team of industry experts performs a line-by-line manual review of the entire contract codebase.
- This meticulous review ensures that no vulnerabilities or flaws go unnoticed, providing an additional layer of scrutiny beyond automated tools.

AI-Assisted Review:

- We leverage advanced AI systems to augment our review process.
- AI algorithms analyze the contract code and provide insights, identifying potential vulnerabilities and recommending improvements.

By combining these rigorous review practices and leveraging cutting-edge tools such as Hardhat, Slither, OpenZeppelin, Solidity Compile, and AI-assisted analysis, SphinxShield delivers a comprehensive assessment of smart contract security. Our approach helps identify and mitigate potential risks, ensuring the solidity codebase is robust, secure, and aligned with industry best practices.

HONEYPOT & RUGPULL SAFETY

Can Take Back Ownership	✅ Not Detected
Owner Change Balance	✅ Not Detected
Blacklist	✅ Not Detected
Modify Fees	🛑 Detected
Personal Modify Fees	✅ Not Detected
Proxy	✅ Not Detected
Whitelisted	✅ Not Detected
Anti Whale	✅ Not Detected
Trading Cooldown	✅ Not Detected
Transfer Pausable	✅ Not Detected
Cannot Sell All	✅ Not Detected
Cannot Buy	✅ Not Detected
Self Destruct	✅ Not Detected
External Call	✅ Not Detected
Hidden Owner	✅ Not Detected
Mint	✅ Not Detected

THREAT LEVELS

At SphinxShield, we categorize vulnerabilities and issues discovered during our smart contract audits into the following threat levels:

- ⚠️ **High Risk:**

  - Issues classified as high risk are critical to the smart contract's performance and functionality.
  - These vulnerabilities pose significant risks and should be addressed and resolved before deploying the contract to a live environment.
  - Failure to mitigate high-risk issues can potentially lead to severe consequences and financial losses.

- ⚠️**Medium Risk:**

  - Issues categorized as medium risk are also critical to the smart contract's performance and functionality.
  - These vulnerabilities, although not as severe as high-risk issues, still require immediate attention and remediation before deploying the contract.
  - Ignoring medium-risk issues may expose the contract to potential attacks or operational disruptions.

- ⚠️ **Low Risk:**

  - Issues classified as low risk are minor details and warnings that do not pose a significant threat to the smart contract's security or functionality.
  - While low-risk issues may not require immediate fixing, they are still worth considering for improvements and best practices.

- ℹ️ **Informational:**

  - The informational level is focused on providing suggestions for enhancing the efficacy or security of certain features in the contract.
  - These recommendations aim to improve the overall robustness and risk-free factor of the contract, even though they do not directly indicate vulnerabilities.

By categorizing vulnerabilities and issues into these threat levels, SphinxShield ensures that critical security concerns are prioritized and addressed promptly. Our experts thoroughly examine the codebase for known vulnerabilities, logical flaws, and access control issues, protecting projects from potential financial damages resulting from malicious exploits.

FOUND THREATS

⚠️High Risk

No high risk-level threats found in this contract

⚠️Medium Risk

No medium risk-level threats found in this contract

⚠️Low Risk

No low risk-level threats found in this contract

ℹ️ Informational

Detected solc-version Impact Informational

Sol Scanning Line:

Pragma version0.8.19 (#5) necessitates a version too recent to be trusted. Consider deploying with 0.8.18.
solc-0.8.19 is not recommended for deployment

Recommendation:

solc frequently releases new compiler versions. Using an old version prevents access to new Solidity security checks.
We also recommend avoiding complex pragma statement.

ℹ️ Informational

Detected assembly Impact Informational

Sol Scanning Line:

Address._revert(bytes,string) (#333-348) uses assembly
- INLINE ASM (#341-344)

Recommendation:

Do not use evm assembly.

ℹ️ Informational

Detected boolean-equal Impact Informational

Sol Scanning Line:

ObamaInu.excludeFromFee(address) (#1050-1057) compares to a boolean constant:
-require(bool,string)(_isExcludedFromFee[account] != true,The wallet is already excluded!) (#1051)
ObamaInu.includeInFee(address) (#1060-1067) compares to a boolean constant:
-require(bool,string)(_isExcludedFromFee[account] != false,The wallet is already included!) (#1061)

Recommendation:

Detects the comparison to boolean constants.

ℹ️ Informational

Detected low-level-calls Impact Informational

Sol Scanning Line:

Low level call in ObamaInu.transferToAddressETH(address,uint256) (#1109-1115):
- (succ) = recipient.call{value: amount}() (#1113)
Low level call in Address.functionStaticCall(address,bytes,string) (#237-250):
- (success,returndata) = target.staticcall(data) (#242)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (#191-211):
- (success,returndata) = target.call{value: value}(data) (#201-203)
Low level call in Address.sendValue(address,uint256) (#102-113):
- (success) = recipient.call{value: amount}() (#108)
Low level call in ObamaInu.recoverETHfromContract() (#1142-1150):
- (succ) = address(marketingWallet).call{value: ethBalance}() (#1144)
Low level call in Address.functionDelegateCall(address,bytes,string) (#276-289):
- (success,returndata) = target.delegatecall(data) (#281)

Recommendation:

The use of low-level calls is error-prone. Low-level calls do not check for code existence or call success.

ℹ️ Informational

Detected naming-convention Impact Informational

Sol Scanning Line:

Parameter ObamaInu.allowance(address,address)._owner (#895) is not in mixedCase
Parameter ObamaInu.setSellFee(uint256)._sellFee (#1103) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (#500) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (#530) is not in mixedCase
Parameter ObamaInu.setMarketingWallet(address)._marketingWallet (#1091) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (#498) is not in mixedCase
Parameter ObamaInu.setSwapAndLiquifyEnabled(bool)._enabled (#1084) is not in mixedCase
Parameter ObamaInu.recoverTokensFromContract(address,uint256)._tokenAddress (#1154) is not in mixedCase
Variable ObamaInu.WETH (#818) is not in mixedCase
Parameter ObamaInu.setBuyFee(uint256)._buyFee (#1097) is not in mixedCase
Function IUniswapV2Router01.WETH() (#570) is not in mixedCase
Parameter ObamaInu.setTokensToSwap(uint256)._minimumTokensBeforeSwap (#1071) is not in mixedCase
Parameter ObamaInu.recoverTokensFromContract(address,uint256)._amount (#1155) is not in mixedCase

Recommendation:

Follow the Solidity naming convention.

ℹ️ Informational

Detected unused-state Impact Informational

Sol Scanning Line:

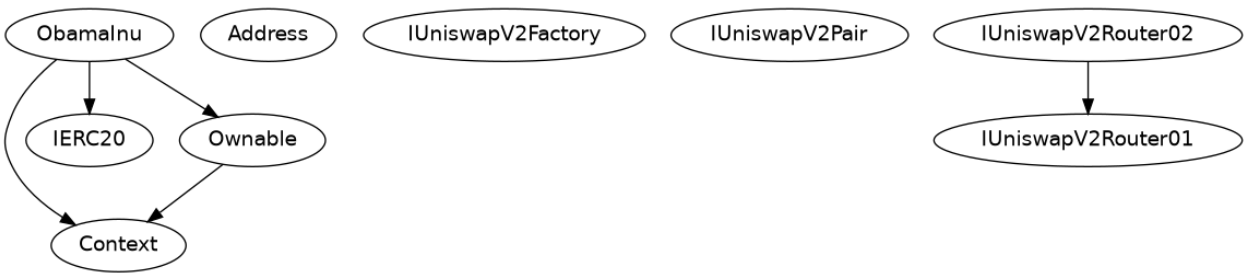
ObamaInu._tFeeTotal (#797) is never used in ObamaInu (#763-1167)

Recommendation:

Remove unused state variables.

CONTRACT GRAPH

CONTRACT INHERITANCE



THE TEAM

⚠️The team hasn't passed any KYC with SphinxShield

KYC INFORMATION

⚠️NO KYC

We encourage the team to undergo a KYC process with SphinxShield to enhance trust and transparency within the community. Please note that our analysis does not consider the KYC status of other companies.

WEBSITE

Website URL
https://obamainu.com/

Domain Registrar
NameCheap, Inc.

Domain Expiration
02 June 2024

Security Tests
Passed. SSL certificate present.

DESIGN
Sleek and visually appealing single-page design incorporating a well-suited color scheme and captivating graphics.

CONTENT
The information immediately conveys the product's purpose to new investors, with no apparent grammar errors.

Whitepaper
Well-crafted and informative, providing clear explanations.

Roadmap
Yes, the goals are defined with specific timeframes.

Mobile-Friendly?
YES

www.ObamaInu.com

SOCIAL MEDIA & ONLINE PRESENCE

ANALYSIS

The project's social media pages demonstrate active engagement.

TWITTER
N/A

TELEGRAM
@ObamaInu

DISCORD
N/A

- 3131 Members
- Active team

ABOUT US

Security Audits | KYC | SAFU
Bug Bounty Campaigns
Vulnerability Scanning
Community-Driven Security
Secure Smart Contract Development
Blockchain Security Consulting

We are a dynamic and fast-growing crypto security agency, specializing in audits, KYCs (Know Your Customer), and consulting services. Our expertise is trusted by some of the most prominent names in the crypto industry.

As a trusted partner to leading crypto projects, we prioritize security and transparency, ensuring that our clients' platforms are robust and protected from potential threats.

Our team of experts excels in conducting thorough audits, scrutinizing code, and assessing smart contracts to identify any vulnerabilities or weaknesses. We provide actionable recommendations to enhance security and mitigate risks.

Partner with our growing crypto security agency to fortify your project's defenses, achieve compliance, and propel your growth in the crypto industry.

Strengthen your crypto project. Hire us now at https://sphinxshield.com/

FIND US ONLINE

TELEGRAM
@SphinxShield

TWITTER
@SphinxShield

DISCLAIMER

The following audit report presents findings based on our assessment of the project at the time of analysis. It covers cybersecurity vulnerabilities, smart contract frameworks, algorithms, and relevant information provided by the project team. While we strive for accuracy and thoroughness, it's important to note that this report is not exhaustive and should not be the sole basis for decision-making.

We strongly advise readers to review the entire report to gain a comprehensive understanding of our analysis. Furthermore, please be aware of the following:

1. Independent Assessment: This report represents our independent assessment and does not constitute financial or investment advice. It is essential to conduct your own research and seek professional advice before making any investment decisions.
2. Limited Scope: Our analysis is based on the information made available to us at the time of the audit. We do not guarantee the absence of undisclosed vulnerabilities or future developments that may impact the project's security.
3. Reliance Limitations: While we have taken utmost care in preparing this report, we cannot be held liable for any reliance placed on its contents. We disclaim any warranties or representations, whether expressed or implied, regarding the accuracy, completeness, or reliability of the information provided.
4. No Endorsement: The inclusion of a project in our audit reports does not constitute an endorsement or guarantee of its security or viability. Investors should exercise their own judgment and conduct thorough due diligence.
5. Non-Transferability: This report is intended solely for the recipient and should not be shared, reproduced, or distributed without our prior written consent.

SphinxShield and its affiliates, including its employees and representatives, disclaim any liability for direct, indirect, incidental, consequential, or other damages arising from the use or reliance on this report, whether in contract, tort, or otherwise.

By accessing and reading this report, you acknowledge and agree to the terms of this disclaimer.

Please proceed with caution and consult appropriate professionals before making any investment decisions.