Smart Contract Audit Report

Solidity Guard

Introduction

This report presents the results of our audit of the DogeEatDoge smart contract. The audit was conducted to assess the overall security, complexity, and best practices employed in the development of the contract.

Review

Contract Name	DogeEatDoge
Compiler Version	v0.8.0
Optimized	No
Verified	Yes
Address	0xe3fcA919883950c5cD468156392a6477Ff5d18de
Network	DogeChain

Scope

Our audit focused on the following areas:

- Codebase Security
- Codebase Complexity and Practices
- Owner privileges and control

Audit Score

Codebase Security	9.5/10
Codebase Complexity and Practices	9/10
Owner privileges and control	10/10
Overall Score	9.5/10

Codebase Security

The codebase security score for this contract is very high, as it follows best practices for secure coding, such as using the latest version of Solidity, using OpenZeppelin libraries, and implementing standard ERC20 functions. However, we recommend adding additional security measures to protect against potential attacks such as adding a delay function for the transfer method to prevent front running attacks.

Codebase Complexity and Practices

The codebase complexity and practices score for this contract is high, as it uses clean and readable code, and follows best practices such as using safe math functions to avoid integer overflow and underflow, and using modifiers to restrict access to functions.

Owner privileges and control

The owner privileges and control of the contract are excellent. The contract does not define any additional functions, which means that the owner has no special privileges or control over the contract's functionality. The contract simply defines a new ERC20 token, which can be transferred like any other ERC20 token.

Conclusion

Overall, we are highly satisfied with the security, complexity, and best practices employed in the development of the DogeEatDoge smart contract. The codebase security and complexity and practices scores are high, indicating that the contract is well-developed and secure. The owner privileges and control score is also excellent, ensuring that the owner has no special privileges or control over the contract's functionality.

However, we recommend adding additional security measures to protect against potential attacks, such as adding a delay function for the transfer method to prevent front running attacks. With this addition, we believe that the DogeEatDoge smart contract will be even more secure and provide a safer environment for its users.

Based on our audit, we give the DogeEatDoge smart contract an overall audit score of 9.5 out of 10.