Smart Contract Audit Report

Executive Summary

The smart contract audited is titled "Deployer Ai" with symbol "devAi" from the Deployer Bot on the Blast Network. Deployer Bot advertises as a service where users only pay gas fees without hidden charges.

Contract Overview

The Deployer Ai contract is a standard implementation of the ERC20 interface with additional customized features for fee management and Uniswap V2 liquidity interactions. The contract includes an Ownable pattern allowing certain functions to be restricted to the contract owner only, such as removing transfer limits or manipulating taxes.
Key characteristics include:

> The contract's total supply is 100,000,000 tokens with 9 decimal places.
> The contract includes a mechanism for fees on transfers, with different rates for buys and sells.
> A maximum wallet size is enforced, initially 1/50 of the total supply per wallet.
> An automatic mechanism for swapping collected tokens for ETH and distributing them to predefined wallets when specific conditions are met.

Functions Overview

The contract comprises a set of functions allowing basic token interactions such as `transfer`, `approve`, and `transferFrom`, as well as owner-only functions such as `openTrading`, `setTax`, `setMaxWallet`, `removeLimits`, `manualSend`, and `manualSwap`.

Detailed Findings

Security

Centralization Risks

Owner Privileges: The contract owner has substantial control over the contract, able to set taxes, remove wallet size limitations, and open trading. This poses centralization risks.
Ownership Renouncement: The contract provides a function to renounce ownership, which, if used, can reduce centralization risk.

Access Controls

The contract uses the modifier `onlyOwner` to restrict certain critical functions to the owner.

Transfer Limits

Max Wallet Size: The contract enforces a maximum wallet size, to prevent wallet addresses from holding more than a specified amount of the token.

Swap Mechanism and Fees

Buy/Sell Tax: The contract establishes a tax on buys and sells, defaulted to 15% for buys and 20% for sells.
Automatic Swapping: The contract can convert tokens automatically into ETH for predefined wallets associated with tax and development funds.
Fee Distribution: ETH is split between a tax wallet (70%) and a dev wallet (30%).

Potential Issues

Slippage Settings: The contract does not allow custom slippage settings which might result in failed transactions during periods of high volatility.
Unset Swap Thresholds: Swap thresholds are hardcoded without allowing dynamic configuration, possibly leading to undesired swapping behavior.

Swap & Liquify: The function `swapTokensForEth` could be called by other functions than anticipated due to its `private` modifier, though it appears to be safe given the current code context.

Miscellaneous

Uniswap Dependency: The contract assumes the existence of a Uniswap V2 Router and a compatible WETH token. Any changes in these dependencies could affect contract behaviors.
Token Clearance: The function `clearStuckToken` allows the contract owner to move any ERC20 tokens sent to the contract.

Recommendations

Decentralization: Implement measures to limit owner privileges to enhance trust in the contract's operations.
Fee Management: Introduce dynamic slippage control and variable management for swap thresholds.
Transparency: Provide clear communication to users about the fee structure and the maximum wallet size constraint.
Testing: Ensure thorough testing, especially of the swap and fee distribution functionalities.

Conclusion

The Deployer Ai smart contract displays standard token functionalities with mechanisms for fees and liquidity generation. The owner has significant control over the contract, which is a point of potential concern for users seeking decentralized solutions. It is recommended to apply best practices to further strengthen the security and decentralization aspects of the contract.
