Respect Token Audit Report

10.24.2024

Overview

The auditing division of TokenMinds consists of a highly experienced team of auditors and developers specializing in securing and optimizing smart contracts on all EVM-based chains.

Our team is dedicated to delivering thorough and precise audits, putting in 100% effort to ensure that each contract is as secure and efficient as possible. We follow a structured four-phase testing process, starting with a manual review to analyze the code for potential issues, followed by functional testing to ensure correct operation. We then conduct security testing to identify vulnerabilities, and finally, perform fuzz testing with randomized inputs to uncover hidden bugs. This comprehensive approach enables TokenMinds to provide a strong, secure foundation for blockchain projects.

Table of Contents

1. Disclaimer

At TokenMinds, our audit process is designed to enhance the quality and security of smart contracts, helping to reduce the risks associated with cryptographic tokens and blockchain technology. However, no audit can guarantee the complete absence of bugs or vulnerabilities. Blockchain technology inherently presents a high level of ongoing risk, and while we aim to identify and mitigate as many issues as possible, absolute security cannot be assured.

This report should not be relied upon for investment decisions or considered as investment advice. Each company or individual is responsible for conducting their own due diligence and maintaining continuous security measures. TokenMinds does not provide any warranties or guarantees regarding the security or functionality of the technology we review, and we encourage follow-up reviews, bug bounty programs, and ongoing monitoring for the best security practices

.

2. Project Overview

Project Name	RESPECT TOKEN
Website	https://respecttoken.com/
Protocol Type	ERC 20
Chain	Ethereum
Language	Solidity
Deployment	https://sepolia.etherscan.io/token/0xc2cf132fbc1f6c99780e745d3b29e28dfa98f756#code
SHA-1 Commit	a349b0131b4d585c2d333ff9b9ae5680411e86ba

3. Risk Classification

3.1. Impact

3.1.1. **High:** Leads to a major loss in assets of protocol or affect large group of users

3.1.2. **Medium:** Little amount of funds can be lost or a core functionality of protocol is only affected

3.1.3. **Low:** Unexpected behavior in protocol without any loss in finances or some non critical protocol functions.

3.2. Likelihood

3.2.1. **High:** Attack path is very likely to be possible with lost of asset outweighing the cost of attack

3.2.2. **Medium:** Conditionally Incentivised attack with high likelihood

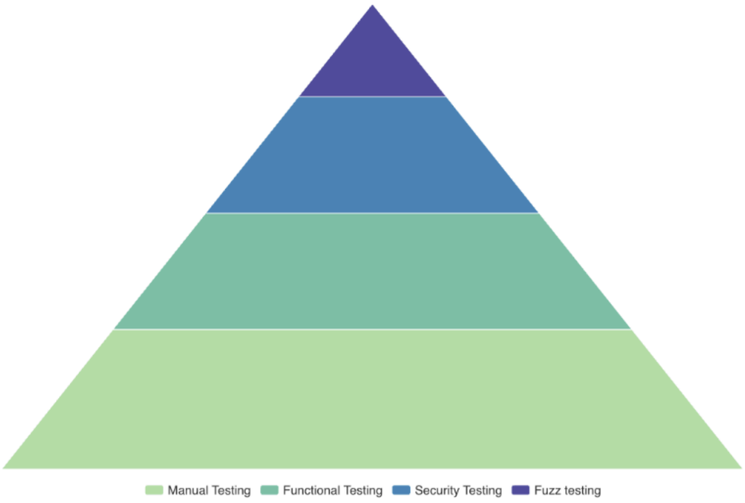
3.2.3. **Low:** The cost of attack may outweigh the lost assets with low likelihood of happening

Severity	**Impact: High**	**Impact: Medium**	**Impact: Low**
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4. Auditing Process

TokenMinds conducts a comprehensive four-level testing process for smart contracts, structured in a pyramid approach. The process begins with a meticulous manual review of the smart contract, ensuring that the code follows best practices and is free from apparent errors.

This is followed by functional testing, where the contract's behavior is thoroughly checked to confirm that it performs its intended functions accurately. Next, the smart contract undergoes rigorous security testing, identifying and addressing potential vulnerabilities that could be exploited. Finally, TokenMinds concludes the process with fuzz testing, which stress-tests the core functionalities by inputting random or unexpected data to ensure stability and robustness under diverse conditions.



This multi-tiered approach ensures a well-rounded assessment of the smart contract's security and functionality.

.

5. Methodology

Our comprehensive audit methodology encompasses several key phases:

5.1. Source Code Examination

5.1.1. Evaluating provided documentation, technical specifications, and implementation guidelines to fully grasp the smart contract's architecture, complexity, and intended functionality

5.1.2. Conducting thorough line-by-line code analysis to identify potential security weaknesses and vulnerabilities

5.1.3. Cross-referencing implementation against provided specifications to verify accurate functionality alignment

5.2. Technical Assessment and Analysis

5.2.1. Implementing test coverage evaluation to measure code execution effectiveness and verify comprehensive testing parameters

5.2.2. Performing dynamic analysis to evaluate contract behavior under various input conditions and execution paths

5.3. Industry Standard Compliance

5.3.1. Assessing contract implementation against established blockchain security patterns

5.3.2. Evaluating code structure for optimization opportunities regarding gas efficiency, readability, and long-term maintenance

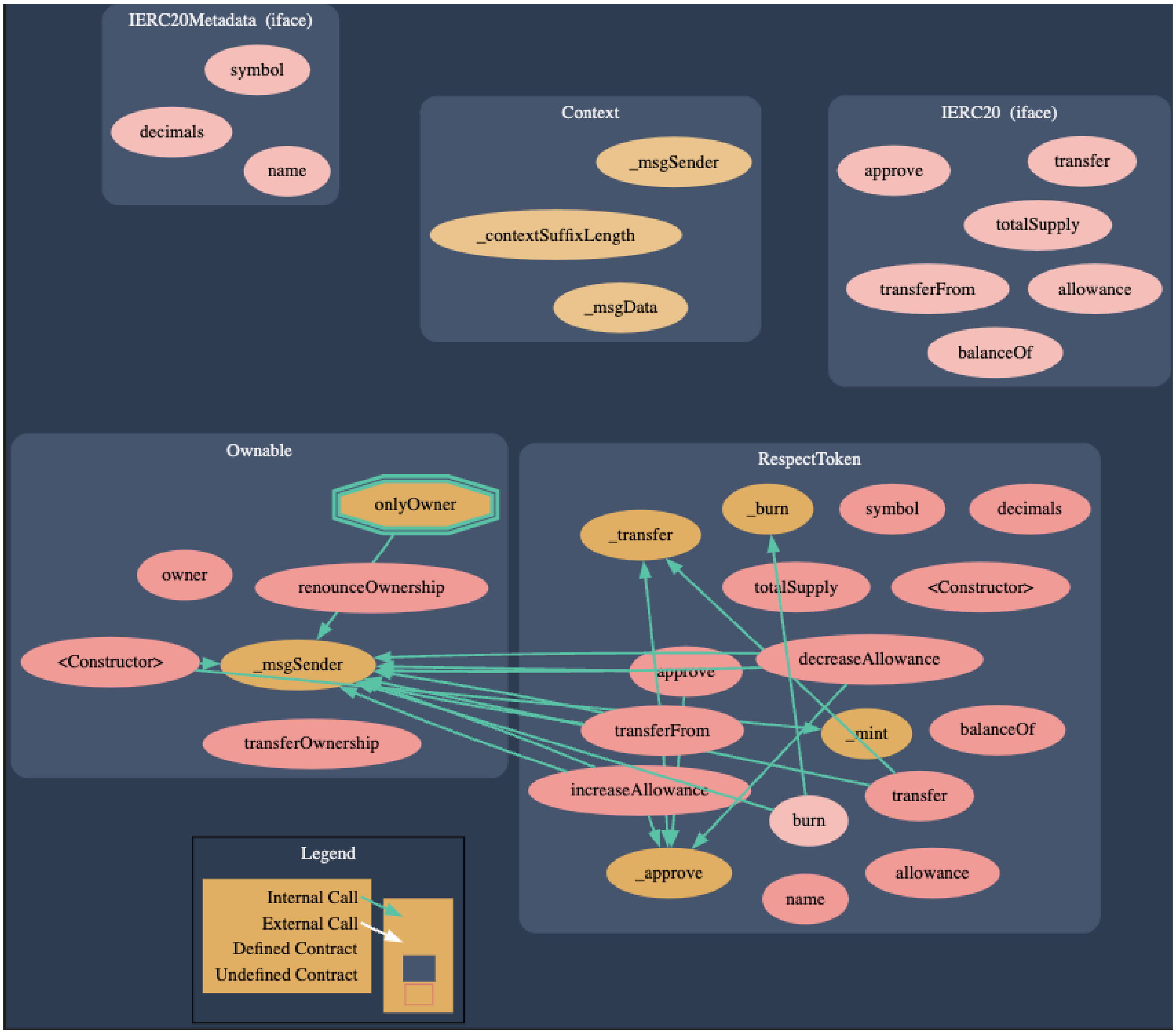
5.3.3. Incorporating latest security insights from both industry practitioners and academic research

5.4. Detailed Remediation Guidelines

5.4.1. Providing specific, implementable security enhancement recommendations

5.4.2. Outlining clear steps for vulnerability mitigation and overall contract strengthening

6. Graphical Representation of RESPECT TOKEN



7. Audit Executive Summary

Audit Version

Audit Version	Delivery Date
v.1.0	24th October, 2024

Commit hash, Code base deployment and disclaimer

Filename	**Commit Hash**
Respect-token/smart-contract/Contract.sol	a349b0131b4d585c2d333ff9b9ae5680411e86ba

Contract address: 0xc2Cf132Fbc1F6c99780E745D3b29E28dfa98f756

Disclaimer: *Files with a different hash value or contract address than those listed above have been modified after the audit by TokenMinds. TokenMinds is not responsible for auditing any files other than those specified above*

Total Function counts in Smart contract

Function Type	Counts
Read-only Functions	8
Write Function	8
Internal Modifier Functions	7
External Modifier Functions	10
Public Modifier Functions	15
Private Modifier Functions	7

Finding Count

Severity	Amount
Critical	0
Medium	0
Low	0
Total Findings	**0**

After an extensive and intensive round of testing, TokenMinds reported ZERO BUGS in the overall contract, ensuring it is ready for secure mainnet deployment. ✅

8. Overall Security - ZERO Bugs ✅

After undergoing four rounds of rigorous testing, the RESPECT Token smart contract has been confirmed to be free of any bugs or vulnerabilities, demonstrating a high level of security and reliability. The comprehensive tests involved stress testing all key functions, including but not limited to:

- **approve**: Allows a user to delegate token spending rights to another account.
- **increaseAllowance**: Increases the token allowance granted to a third party.
- **decreaseAllowance**: Reduces the token allowance of a third party.
- **transfer**: Facilitates the transfer of tokens between two parties.
- **transferFrom**: Enables a third-party account to transfer tokens on behalf of the owner, within the allowed limit.
- **burn**: Permits token holders to destroy tokens, reducing the total supply.

Each of these functions has been thoroughly validated, and no security flaws or issues were detected during the testing phases. This confirms that the RESPECT Token contract is secure, reliable, and ready for deployment without concerns of exploitation or malfunction.

Invariant Testing and Functional Testing with Echidna and Foundry

Function Name	Pass/Fail 🟩 🟥	Optimization Remark
transfer()	✅	Optimized
approve()	✅	Optimized
increaseAllowance()	✅	Optimized
decreaseAllowance()	✅	Optimized
transferFrom()	✅	Optimized
burn()	✅	Optimized
Internal _transfer()	✅	Optimized

Other Checks

Checks	Description	Presence	Remarks
Upgradeability	A contract that can be upgraded to modify their code, while preserving their state, address and balance	❎	This contract cannot be upgraded
Mints	Ability to mint new tokens	❎	The contract mints 110,000,000,000 at the time of deployment and cannot mint any tokens further.
Burn	The owner should have the ability to burn tokens without any allowance	✅	N/A
Blacklist	Contract owner cannot blacklist addresses from owning the token	✅	There is no such function that blacklist any wallet addresses
Pausable	The contract cannot be paused	✅	No function to pause the contract
Centralized Privileges	Centralized Privileges	❎	No centralized privileges found
Withdraw	Withdraw function to transfer the funds from contract to a set of addresses	❎	No withdraw function present (Not needed in an ERC20 contract)

9. Conclusion

TokenMinds has concluded the audit of the RESPECT Token. Following a comprehensive review, no bugs or vulnerabilities were found in the smart contract. The contract exhibits a highly optimized structure, adhering to best practices in both security and efficiency. The audit included a detailed manual code analysis, functional and security testing, as well as fuzz testing to ensure the contract's robustness.