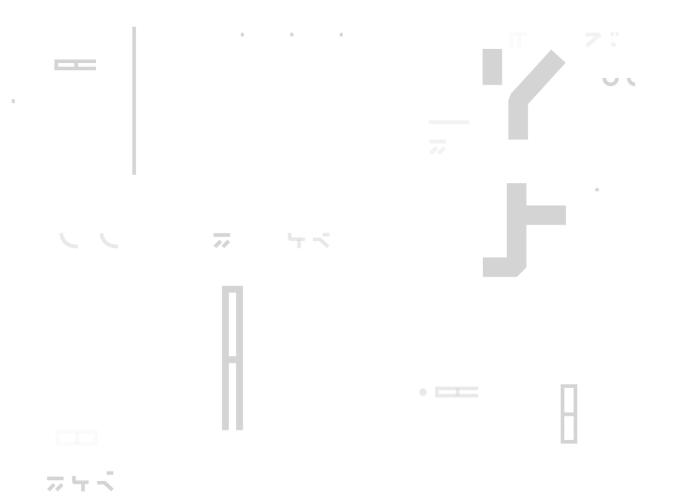
HACKEN

Ч

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Customer: Tr3zor.io Date: November 19th, 2021



This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed – upon a decision of the Customer.

Document

Name	Smart Contract Code Review and Security Analysis Report for Tr3zor.io.		
Approved by	Andrew Matiukhin CTO Hacken OU		
Туре	ERC20 token		
Platform	Ethereum / Solidity		
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review		
Deployed contract	https://etherscan.io/address/0xF61BF4d1A948487d61b8fa63808AAC06 bda55F98#code		
Technical Documentation	YES		
JS tests	NO		
Website	tr3zor.io		
Timeline	17 NOVEMBER 2021 - 19 NOVEMBER 2021		
Changelog	19 NOVEMBER 2021 - INITIAL AUDIT		



Table of contents

Introduction	4
Scope	4
Executive Summary	5
Severity Definitions	6
Audit overview	7
Conclusion	8
Disclaimers	9



Introduction

Hacken OÜ (Consultant) was contracted by Tr3zor.io (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contract and its code review conducted between November 17th, 2021 - November 19th, 2021.

Scope

```
The scope of the project is smart contracts in the blockchain:

URL:

https://etherscan.io/address/0xF61BF4d1A948487d61b8fa63808AAC06bda55F

98#code

Technical Documentation: Yes (Whitepaper:

https://tr3zor.io/uploads/Tr3zor_WP-en.pdf)

JS tests: No

Contracts:

TR3.sol

Context.sol

ERC20.sol

IERC20.sol

SafeMath.sol
```

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item	
Code review	 Reentrancy 	
	 Ownership Takeover 	
	 Timestamp Dependence 	
	 Gas Limit and Loops 	
	 DoS with (Unexpected) Throw 	
	 DoS with Block Gas Limit 	
	 Transaction-Ordering Dependence 	
	 Style guide violation 	
	 Costly Loop 	
	 ERC20 API violation 	
	 Unchecked external call 	
	 Unchecked math 	
	 Unsafe type inference 	
	 Implicit visibility level 	
	 Deployment Consistency 	
	 Repository Consistency 	
	 Data Consistency 	

<u>www.hacken.io</u>

	L	Ц	Ц
Ë		1	П
Ш			

Functional review	 Business Logics Review
	 Functionality Checks
	 Access Control & Authorization
	 Escrow manipulation
	 Token Supply manipulation
	 Assets integrity
	 User Balances manipulation
	 Data Consistency manipulation
	 Kill-Switch Mechanism
	 Operation Trails & Event Generation

Executive Summary

According to the assessment, the Customer's smart contracts are well-secured.

Insecure	Poor secured	Secured	Well-secured
		You are here	

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview section.

As a result of the audit, security engineers found **no** issues.



Severity Definitions

Risk Level	Description	
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.	
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions	
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.	
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution	



Audit overview

🛛 🗖 🗖 Critical

No critical issues were found.

📕 📕 📕 High

No high severity issues were found.

🔳 🔳 Medium

No medium severity issues were found.

Low

No low severity issues were found.



Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

As a result of the audit, security engineers found **no** issues.



Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only – we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.