

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Customer: Cryptia

Date: November 24th, 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 Cryptia.		
Approved by	Andrew Matiukhin CTO Hacken OU		
Туре	BEP20 token with Fees		
Platform	Binance Smart Chain / Solidity		
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review		
Deployed	https://bscscan.com/address/0x142ba459218cafdcf8f98a5fa3e139f32		
contract	fccec70#code		
Technical	NO		
Documentation			
JS tests	NO		
Website	cryptia.io		
Timeline	22 NOVEMBER 2021 - 24 NOVEMBER 2021		
Changelog	24 NOVEMBER 2021 - INITIAL AUDIT		

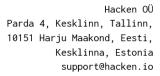




Table of contents

Introduction	
Scope	4
Executive Summary	5
Severity Definitions	7
Audit overview	8
Conclusion	9
Disclaimers	10



Introduction

Hacken OÜ (Consultant) was contracted by Cryptia (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 22^{nd} , 2021 - November 24^{th} , 2021.

Scope

```
The scope of the project is smart contracts deployed:
      https://bscscan.com/address/0x142ba459218cafdcf8f98a5fa3e139f32fccec7
0#code
Technical Documentation: No
JS tests: No
Contracts:
      CryptToken.sol
      ERC20.sol
      ERC20Capped.sol
      ERC20Burnable.sol
      Address.sol
      Ownable.sol
      SafeMath.sol
      IERC20.sol
      IERC20Metadata.sol
      Context.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	



	• Democitany Consistency
	Repository Consistency
	Data Consistency
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 1 low severity issue.

Notice:



Token takes 5% fees of all transactions done to the address listed in the "_exchangeWallets" mapping. This mapping is controlled by the owner.



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

Too many digits.

Literals with many digits are difficult to read and review.

Function: constructor

Recommendation: While it's already deployed, nothing needs to be changed. But for the future, we'd recommend using scientific notation with ether units suffix. For example: 500e6 ether



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 1 low severity issue.

Notice:

Token takes 5% fees of all transactions done to the address listed in the "_exchangeWallets" mapping. This mapping is controlled by the owner.



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.