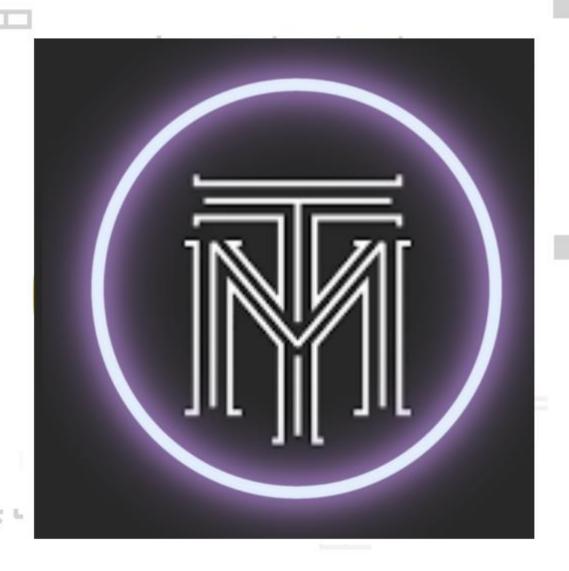


# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Customer: TRADEMATE

Date: June 27<sup>th</sup>, 2023



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 fixed - upon a decision of the Customer.

#### Document

Name	Smart Contract Code Review and Security Analysis Report for CEOS.		
Approved	Andrew Matiukhin   CTO Hacken OU		
by			
Туре	Token		
Platform	Binance Smart Cahin / Solidity		
Methods	Architecture Review, Functional Testing, Computer-Aided		
	Verification, Manual Review		
Deployed	https://bscscan.com/address/0x46421354e9f6bf42777c6f83dae557687c8f5		
contract	<u>702</u>		
Timeline	27 JUN 2023 - 27 JUN 2023		
Changelo	27 JUN 2023 - INITIAL AUDIT		
g			

#### Table of contents

Scope	3
Executive Summary	4
Severity Definitions	-
Conclusion	7
Digalaimana	c



#### Introduction

Hacken OÜ (Consultant) was contracted by Titan Hunters (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer's smart contract and its code review conducted between JUN  $25^{\rm th}$ , 2023 – JUN  $27^{\rm th}$ , 2023.

#### Scope

The scope of the project is smart contracts in the repository:

Contract deployment address:

https://bscscan.com/address/0x46421354e9f6bf42777c6f83dae557687c8f 5702 File:

Trademate.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:

Vainciabilicies	that are considered:
Category	Check Item
Code review	Reentrancy
	<ul><li>Ownership Takeover</li></ul>
	Timestamp Dependence
	Gas Limit and Loops
	DoS with (Unexpected) Throw
	DoS with Block Gas Limit
	Transaction-Ordering Dependence
	Style guide violation
	<ul><li>Costly Loop</li></ul>
	ERC20 API violation
	<ul><li>Unchecked external call</li></ul>
	Unchecked math
	• Unsafe type inference
	Implicit visibility level
	Deployment 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
	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	

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. A general overview is presented in AS-IS section, and all found issues can be found in the Audit overview section. Security engineers found no issues during the audit.

Notice: the audit scope is limited and does not include all files in the repository. Though, reviewed contracts are secured, we may not guarantee secureness of contracts that are not in the scope.



# Severity Definitions

Risk Level	Description		
	Critical vulnerabilities are usually		
Critical	straightforward to exploit and can lead to		
	assets loss or data manipulations.		
	High-level vulnerabilities are difficult to		
High	exploit; however, they also have a significant		
111911	impact on smart contract execution, e.g.,		
	public access to crucial functions		
	Medium-level vulnerabilities are important to		
Medium	fix; however, they can't lead to assets loss or		
	data manipulations.		
	Low-level vulnerabilities are mostly related		
Low	to outdated, unused, etc. code snippets that		
	can't have a significant impact on execution		
Lowest /	Lowest-level vulnerabilities, code style		
Code	violations, and info statements can't affect		
Style /	smart contract execution and can be ignored.		
Best			
Practice			



### Audit overview

#### ■ ■ ■ Critical

No critical issues were found.

# High

No high severity issues were found.

#### ■ ■ Medium

No medium issues were found.

#### Low

No low issues were found.



## Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For the contract, high-level description of functionality was presented in As-Is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code. Security engineers found no issue during the audit.

Notice: the audit scope is limited and does not include all files in the repository. Though, reviewed contracts are secured, we may not guarantee secureness of contracts that are not in the scope.



#### **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 security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree 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 security of smart contracts. **Technical Disclaimer** 

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