HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Asyagro Tech Solutions LLC
Date: March 30th, 2022

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Document

Name	Smart Contract Code Review and Security Analysis Report for Asyagro Tech Solutions LLC.
Approved By	Evgeniy Bezuglyi | SC Department Head at Hacken OU
Type of Contracts	ERC20 token; Locking
Platform	EVM
Language	Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Website	https://www.asyagro.io/
Timeline	01.03.2022 - 30.03.2022
Changelog	03.03.2022 - Initial Review 30.03.2022 - Revision

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Table of contents

Introduction 4

Scope 4

Executive Summary 5

Severity Definitions 7

Audit overview 8

Recommendations 10

Disclaimers 11

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Introduction

Hacken OÜ (Consultant) was contracted by Asyagro Tech Solutions LLC (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 contracts.

Scope

The scope of the audit is deployed smart contracts:

Technical Documentation: Yes
JS tests: No
Contracts code:
https://bscscan.com/address/0xc0cc1e5761ba5786916fd055562551798e50d573#code

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

Repository:
https://github.com/Asyagro/ASY
Commit:
7ed2ad0ba2de200ace36f9e21eb0ebcaa1e343ff
Technical Documentation: Yes
JS tests: No
Contracts:
./asyagroV2.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 ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ EIP standards violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency

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

Executive Summary

The score measurements details can be found in the corresponding section of the methodology.

Documentation quality

The Customer provided superficial functional requirements and technical requirements. Total Documentation Quality score is **6** out of **10**.

Code quality

The total CodeQuality score is **5** out of **10**. No unit tests were provided.

Architecture quality

The architecture quality score is **5** out of **10**. It is better to block tokens in a separate contract so that the token contract will be clean.

Security score

As a result of the audit, security engineers found **4** critical and **1** medium severity issues.

As a result of the revision, security engineers found **1** **new** high severity issue. All previously found issues were fixed.

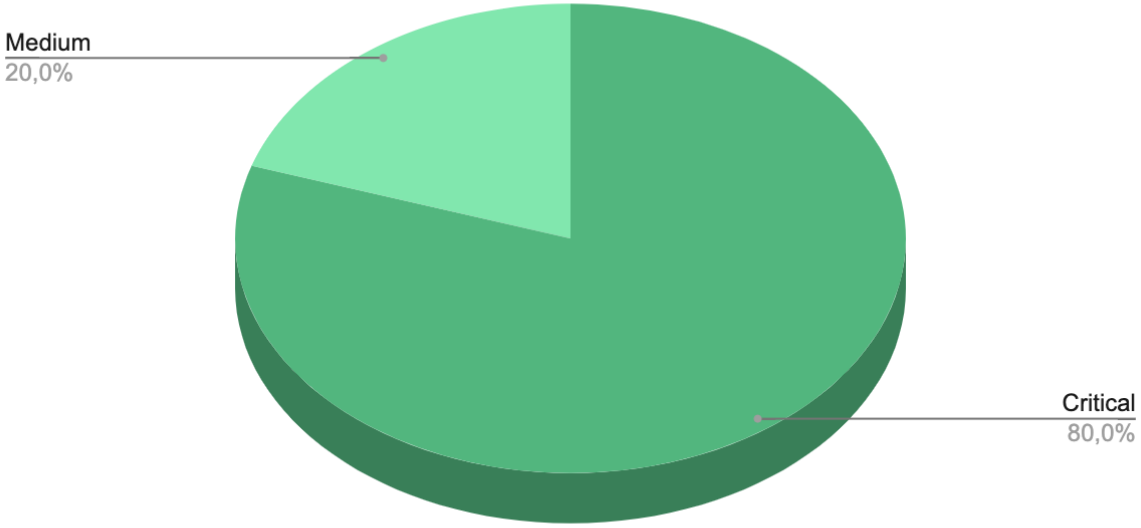
The security score is **5** out of **10**. All found issues are displayed in the “Issues overview” section.

Summary

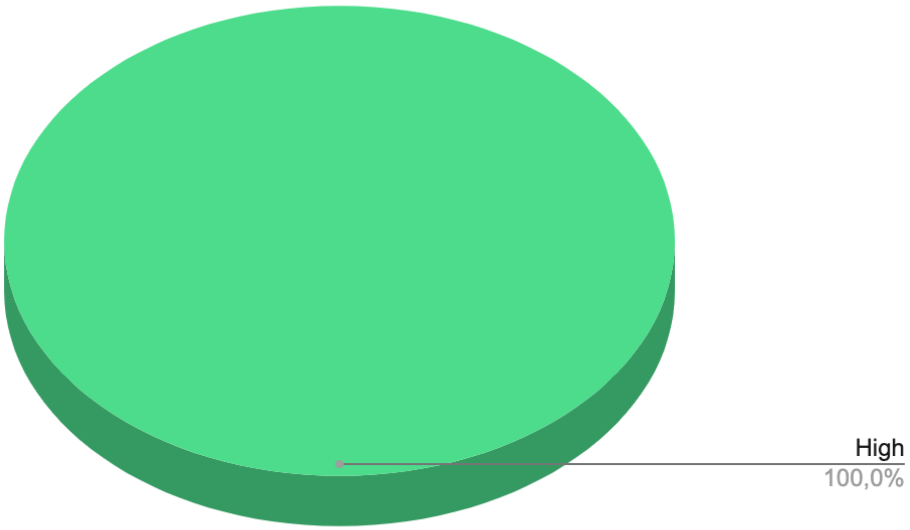
According to the assessment, the Customer's smart contracts has the following score: **5.1**

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Graph 1. The distribution of vulnerabilities after the audit.



Graph 2. The distribution of vulnerabilities after the revision.



Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

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 cannot lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that cannot have a significant impact on execution

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Audit overview

■■■■ Critical

1. Owners can mint tokens.

   According to the tokenomics maximum total supply is 7,500,000,000 ASY, but owners can mint more tokens using the mint function.

   Contracts: ASYAGRO

   Function: mint

   Recommendation: remove the ability to mint more than stated in tokenomics.

   Status: Fixed (Revised commit: 7ed2ad0)

2. Owners can lock all tokens of any user anytime.

   Lock functionality should be limited by clear contract rules. Owners should not be able to block user tokens at their discretion.

   Contracts: ASYAGRO

   Function: lock

   Recommendation: change lock functionality.

   Status: Fixed (Revised commit: 7ed2ad0)

3. Owners can change the lock time after the lock is created.

   The ability to change the lock time of an already created lock can lead to various manipulations.

   Contracts: ASYAGRO

   Functions: extendLockTime, reduceLockTime

   Recommendation: remove the ability to change the lock time after the lock is created.

   Status: Fixed (Revised commit: 7ed2ad0)

4. Owners can unlock tokens anytime.

   The ability to unlock tokens for any account at any time can lead to various manipulations.

   Contracts: ASYAGRO

   Functions: unlockToken, releaseLock

   Recommendation: remove the ability to unlock tokens before the end of the lock period.

Status: Fixed (Revised commit: 7ed2ad0)

High

1. Highly permissive owner access.

   Owners can add the user's address to the 'frozen' list. All ASY token transfers from such addresses will be reverted.

   This can lead to various manipulations and even loss of funds by users.

   Contracts: ASYAGRO

   Function: freezeAccount

   Recommendation: remove the possibility to block the user’s funds.

   Status: New

Medium

1. Unused function.

   The freezeAccount function does nothing.

   Contracts: ASYAGRO

   Function: freezeAccount

   Recommendation: remove unused code.

   Status: Fixed (Revised commit: 7ed2ad0)

Low

No low severity issues were found.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Recommendations

1. We recommend following the single responsibility principle and moving the locking (vesting) functionality to a separate contract.

   Contracts: ASYAGRO

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with 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 a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements. 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 cannot guarantee the explicit security of the audited smart contracts.