

HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: BotPlanet
Date: April 04th, 2022

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 BotPlanet. |
| Approved By | Evgeniy Bezuglyi SC Department Head at Hacken OU |
| Type of Contracts | DEX Router, Libraries |
| Platform | EVM |
| Language | Solidity |
| Methods | Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review |
| Website | https://www.botpla.net/ |
| Timeline | 16.02.2022 - 04.04.2022 |
| Changelog | 23.03.2022 - Initial Review 04.04.2022 - Revise |



Table of contents

| | |
|----------------------|----|
| Introduction | 4 |
| Scope | 4 |
| Executive Summary | 6 |
| Severity Definitions | 8 |
| Findings | 9 |
| Disclaimers | 10 |

Introduction

Hacken OÜ (Consultant) was contracted by BotPlanet (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 project is smart contracts in the repository:

Repository:

<https://github.com/BOTDeFi/botdex-contracts-periphery>

Commit:

5c8683be30aa5f8c23fbdfe8407de79c2c5c63d4

Technical Documentation: Yes

(<https://www.botpla.net/wp-content/uploads/2022/02/White-Paper.pdf>;
<https://docs.google.com/document/d/1WM0kMi4xAX7MburhZyQ8wy6Axmyf9xXtpnvL15yuNgs/edit>)

JS tests: Yes (test)**Contracts:**

```
interfaces/IBotdexMigrator.sol
interfaces/IBotdexRouter01.sol
interfaces/IBotdexRouter02.sol
interfaces/IWETH.sol
interfaces/external/IBotdexCallee.sol
interfaces/external/IBotdexERC20.sol
interfaces/external/IBotdexFactory.sol
interfaces/external/IBotdexPair.sol
interfaces/external/IERC20.sol
interfaces/V1/IUniswapV1Exchange.sol
interfaces/V1/IUniswapV1Factory.sol
libraries/Babylonian.sol
libraries/BotdexLibrary.sol
libraries/BotdexOracleLibrary.sol
libraries/FixedPoint.sol
libraries/SafeMath.sol
libraries/TransferHelper.sol
test/ERC20.sol
test/DeflatingERC20.sol
test/RouterEventEmitter.sol
test/WETH9.sol
BotdexMigrator.sol
BotdexRouter.sol
BotdexRouter01.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 | <ul style="list-style-type: none">▪ 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 | <ul style="list-style-type: none">▪ 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

Score measurements details can be found in the corresponding section of the [methodology](#).

Documentation quality

The customer provided a whitepaper as functional requirements and a short description of functions as technical requirements. Counting that the implementation is a 1-to-1 clone of the UniswapV2-periphery which is fully documented, the total Documentation Quality score is **8** out of **10**.

Code quality

Total CodeQuality score is **8** out of **10**. The code is a copy of the UniswapV2-periphery codebase. Some parts are improved.

Architecture quality

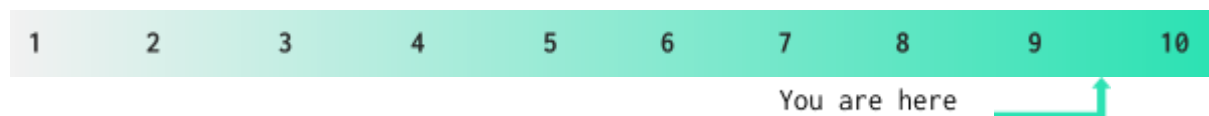
The architecture quality score is **9** out of **10**. The architecture is standard for the DEX periphery.

Security score

As a result of the audit, security engineers found **no security issues**. The security score is **10** out of **10**. All found issues are displayed in the “Issues overview” section of the report.

Summary

According to the assessment, the Customer's smart contract has the following score: **9.5**



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 |

Findings

■■■■ Critical

No critical severity issues were found.

■■■ High

No high severity issues were found.

■■ Medium

No medium severity issues were found.

■ Low

1. Floating solidity version

It is recommended to specify the exact solidity version in the contracts.

Contracts: all

Recommendation: please specify exact solidity version (ex. `pragma solidity 0.8.4` instead of `pragma solidity >=0.8.0`).

Status: Fixed (Revised Commit: 5c8683b)

2. SPDX license identifier not provided in a source file.

Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to each source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source code. Please see <https://spdx.org> for more information.

Contracts: all

Recommendation: add SPDX-license identifiers.

Status: Fixed (Revised Commit: 5c8683b)



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.