北京链安
Chains Guard Technology

# PILOT PROTOCOL
# Smart Contract
# Audit Report

Beijing ChainsGuard Technology

## ■ Documentation

| File name | PILOT PROTOCOL Smart Contract Audit Report | | |
|---|---|---|---|
| Serial number | CG-YMSJ-EN-20210312 | | |
| Confidentiality | Business secret | Version | V1 |
| Author | ChainsGuard<br>Security Center | Date | 2021-03-12 |

## ■ Scope of application

This security assessment is authorized by the project party. Beijing ChainsGuard Network Technology Co., Ltd. (hereinafter referred to as "ChainsGuard") conducts an in-depth security risk assessment of the PILOT PROTOCOL smart contract; the technical report submitted according to the assessment result is used for The security status of the smart contract makes security assessment and reinforcement recommendations. only limited to ChainsGuard and the internal personnel of the project party.

## ■ Version change record

| Date | Version | Description | Modify by |
|---|---|---|---|
| 2021-03-12 | V1 | Document creation | Berry |

# Catalogue

Disclaimer . . . . . 1

1 Introduction . . . . . 2

1.1 Overview . . . . . 2

1.2 Audit Time . . . . . 2

1.3 Audit Unit . . . . . 2

1.4 Audit Object . . . . . 2

2 Security Audit Summary . . . . . 3

2.1 Vulnerability Statistics . . . . . 3

2.2 Audit items . . . . . 4

3 Detailed Results . . . . . 5

3.1 Math Overflow . . . . . 6

3.2 Heco security . . . . . 6

3.3 Contract backdoor . . . . . 6

4 Summary of Security Audit . . . . . 7

Appendix A. Explanation of Security Risk Status Levels . . . . . 8

# Disclaimer

The audit report is a technical security audit for the authorized party. The purpose of this audit is to provide the authorized party with a reference basis for conducting its business security assessment and optimization, The regulatory regime of the business model, or any other statement about the applicability of the application, as well as a statement or warranty that the application is in error-free behavior. This report cannot be used as a proof of that these tested systems and codes are absolutely secure and there are no other security risks.

The audit report only covers the code, installation packages and other materials provided by the authorized party, and its conclusion is only applicable to the corresponding version of the application. Once the relevant code, configuration, and operating environment change, the corresponding conclusion will no longer be applicable.

This audit is limited to technical security audits of the smart contract, but the security of other programs, applications, front-end pages, and other technical modules which invoke this smart contract is not within the scope of the audit. At the same time, non-technical risks such as moral risk, operational risk and market risk arising from the actual use of the smart contract are not related to the technical audit results of this smart contract.

# 1 Introduction

## 1.1 Overview

This document includes the results of the audit performed by the Chains Guard Team on the PILOT PROTOCOL project, at the request of the PILOT PROTOCOL team. The goal of this audit is to review the smart contract cod e solidity implementation, study potential security vulnerabilities, its general designand architecture, and uncover bugs that could compromise the softw are in production.

## 1.2 Audit Time

| Evaluation test time | |
|---|---|
| **Start time** | 2021-03-09 |
| **End time** | 2021-03-12 |

## 1.3 Audit Unit

| | |
|---|---|
| **Company Name** | The Beijing ChainsGuard Network Technology Co., Ltd. |
| **Web Site** | https://www.chainsguard.com/ |

## 1.4 Audit Object

| Name | Contract Md5 | Version |
| --- | --- | --- |
| PILOT PROTOCOL | Bank.sol<br>475a3c14a45ac9ea32245775cf793941<br>MdxStrategyAddTwoSidesOptimal.sol<br>8d3c22a7ce9ee234332665947ca6be44<br>MdxGoblin.sol<br>e5e7ff38d08f8e32275b8f8f61a183ce<br>MdxStrategyWithdrawMinimizeTrading.sol<br>73ce0b4e0fcced7d03a2af7da3a3d708 | V1 |

# 2 Security Audit Summary

## 2.1 Vulnerability Statistics

| Vulnerabilities | High risk | Medium risk | Low risk |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |

**[Note] A brief description of the hazard classification method is as follows**

**High**: It directly causes the system to be controlled or the data to be destroyed. Once it occurs, it is a serious security event.

**Medium**: It may lead to the leakage of important information or may cause the system to be controlled.

**Low**: Non-critical information leaks or minor security issues generally do not lead to serious security incidents.

## 2.2 Audit items

We focus on the review of the following inspection items:

| Attack surface | Check list | status | description |
|---|---|---|---|
| Reentrancy Attack | Cross-contract interaction | Pass | Unprotected sensitive functions call external contracts |
| | HT Transfer | Pass | Unrestricted Gas transfer HT has a hidden danger of reentry |
| Unauthorized access | Constructor does not match | Pass | Whether the contract name and constructor in the lower version do not match |
| | Privileged function exposure | Pass | Exposure of privileged functions caused by incorrect authentication methods |
| | tx.origin variable abuse | Pass | Whether the contract uses tx.origin for identity authentication |
| | Access control flaws | Pass | Unreasonable settings for the visibility of functions and state variables |
| Numerical overflow | Overflow & underflow | Pass | Does the contract have common overflow or underflow vulnerabilities |
| Race condition | Transaction order dependence | Pass | Does the final state of the contract depend on the order of transactions |

| | | | |
|---|---|---|---|
| Denial of service | Unexpected transaction rollback | Pass | Is the contract vulnerable to revert cause denial of service |
| | Gas Price exceeded | Pass | Excessive Gas Price caused by excessive loop |
| call injection | call function abuse | Pass | The contract receives external input as a parameter of the call function |
| Fake recharge | Recharge result check | Pass | Whether the contract uses an incorrect method to check the recharge result |
| Miner privileges | Timestamp dependence | Pass | Does the contract rely on the timestamp to complete the main function |
| | fake-random number dependence | Pass | Does the contract rely on pseudo-random numbers to complete its main functions |
| Heco | Business logic | Pass | shareholders can arbitrarily reward withdrawals |
| Other checks | External input check | Pass | Whether the contract verifies the legality of external input |
| | Use untrusted libraries | Pass | Whether the contract uses untrusted (unsafe) libraries |
| | Leakage of sensitive information | Pass | Does the contract have hidden dangers of leaking sensitive information |
| | Blackhole | Pass | Whether the contract locks HT or tokens indefinitely |
| | Contract backdoor | Pass | Does the contract have a backdoor that can be controlled by the project party |

# 3 Detailed Results

After the cooperation of the project party and the auditor during this au dit, all contracts have been in a state of three security issues.

## 3.1 Math Overflow

The value processing in the contract needs to strictly check the arithmetic overflow problem. The conventional addition and subtraction arithmetic processing is easy to cause integer overflow or underflow, especially when dealing with the account amount of similar tokens, it is necessary to strictly judge the size before and after the account amount. Normally, it is recommended to use OpenZeppelin open source library SafeMath module for numerical calculations.

Audit result: pass

## 3.2 Heco security

Review whether there are obvious flaws in the business logic of Decentralized finance, whether the functions meet business needs, whether there are security risks in digital asset custody, and whether the oracle is implemented and used safely.

Audit result: **pass**

## 3.3 Contract backdoor

In the blockchain ecology, some project parties do not have the ability to develop smart contracts. They usually choose some smart contract automatic generation tools to generate one-click and automatically deploy to the chain. This brings hackers to insert backdoor code into smart contracts. opportunity.

Once the smart contract is inserted into the backdoor code, hackers can use the backdoor to manipulate the contract arbitrarily, which will bring fatal harm to the project party. Or the contract has a backdoor that can be controlled by the project party.

Audit result: pass

# 4 Summary of Security Audit

The overall assessment security status of this audit is: **good status**

The intelligent contract audit results only provide the actual basis for the authorizer to formulate corresponding security measures and solutions.

# Appendix A. Explanation of Security Risk Status Levels

| Security risk status statement | |
|---|---|
| 1 | **good status**<br>The contract is in good running condition, and there are no or only sporadic low-risk security problems. At this time, as long as the existing security policy is maintained, the safety level requirements of the system can be met. |
| 2 | **warning status**<br>There are some loopholes or security risks in the smart contract, which have not been used on a large scale. At this time, targeted reinforcement or improvement should be carried out according to the problems found in the evaluation, and then redeployment. |
| 3 | **serious status**<br>Smart contract has been widely used. Serious loopholes or security problems that may seriously threaten the normal operation of the contract are found in the intelligent contract. At this time, measures should be taken immediately to redeploy the strengthened intelligent contract. |
| 4 | **emergency status**<br>The tokens related to the intelligent contract have been opened for trading. Serious loopholes or security problems that may seriously threaten the normal operation of the contract have been found in the intelligent contract, which may cause serious damage to economic interests. At this point, should immediately stop the contract related token trading, immediately take measures to redeploy the strengthened intelligent contract. |