北京链安
Chains Guard Technology

PizzaSwap Smart Contract Review Report

Beijing ChainsGuard Technology

Documentation

File name	PizzaSwap Smart Contract Review Report		
Serial number	CG-YMSJ-EN-202010033		
Confidentiality	Business secret	Version	V1.1
Author	ChainsGuard Security Center	Date	2020-10-30

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 PizzaSwap 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
2020-10-30	V1.1	Document creation	ChainsGuard Security Center

Catalogue

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 PizzaSwap project, at the request of the PizzaSwap tea m. The goal of this audit is to review the smart contract code solidity impl ementation, study potential security vulnerabilities, its general designand arc hitecture, and uncover bugs that could compromise the software in product ion.

1.2 Audit Time

Evaluation test time	
Start time	2020-10-29
End time	2020-10-30

1.3 Audit Unit

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

1.4 Audit Object

pizzaswap2.sol SHA256：5b6332fccd4b2ae61280566ad5bfeacab649362988877dde6cc2d830bb390dd1

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
	ETH Transfer	Pass	Unrestricted Gas transfer ETH 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
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 ETH or tokens indefinitely

	Contract backdoor	Pass	Does the contract have a backdoor that can be controlled by the project party

3 Detailed Results

3.1 totalAllocPoint Variable Accumulation Bug

In the liquidity mining MasterChef contract, when the contract owner adds a new liquidity pool, the totalAllocPoint variable of the liquidity pool will not accumulate, which will cause deviations in subsequent mining revenue calculations.

```
// Add a new lp to the pool. Can only be called by the owner.
// XXX DO NOT add the same LP token more than once. Rewards will be messed up if you do.
function addLp(uint256 _allocPoint, IERC20 _lpToken) public onlyOwner {
    uint256 lastRewardTime = block.timestamp > starttime ? block.timestamp : starttime;
    //totalAllocPoint = totalAllocPoint.add(_allocPoint);
    poolInfo.push(PoolInfo({
        lpToken: _lpToken,
        allocPoint: _allocPoint,
        lastRewardTime: lastRewardTime,
        accPizzaPerShare: 0,
        totalPool:0
    }));
}
```

Fix suggestion: Just open the comment, if it is a business need or based on other factors, please explain the reason.

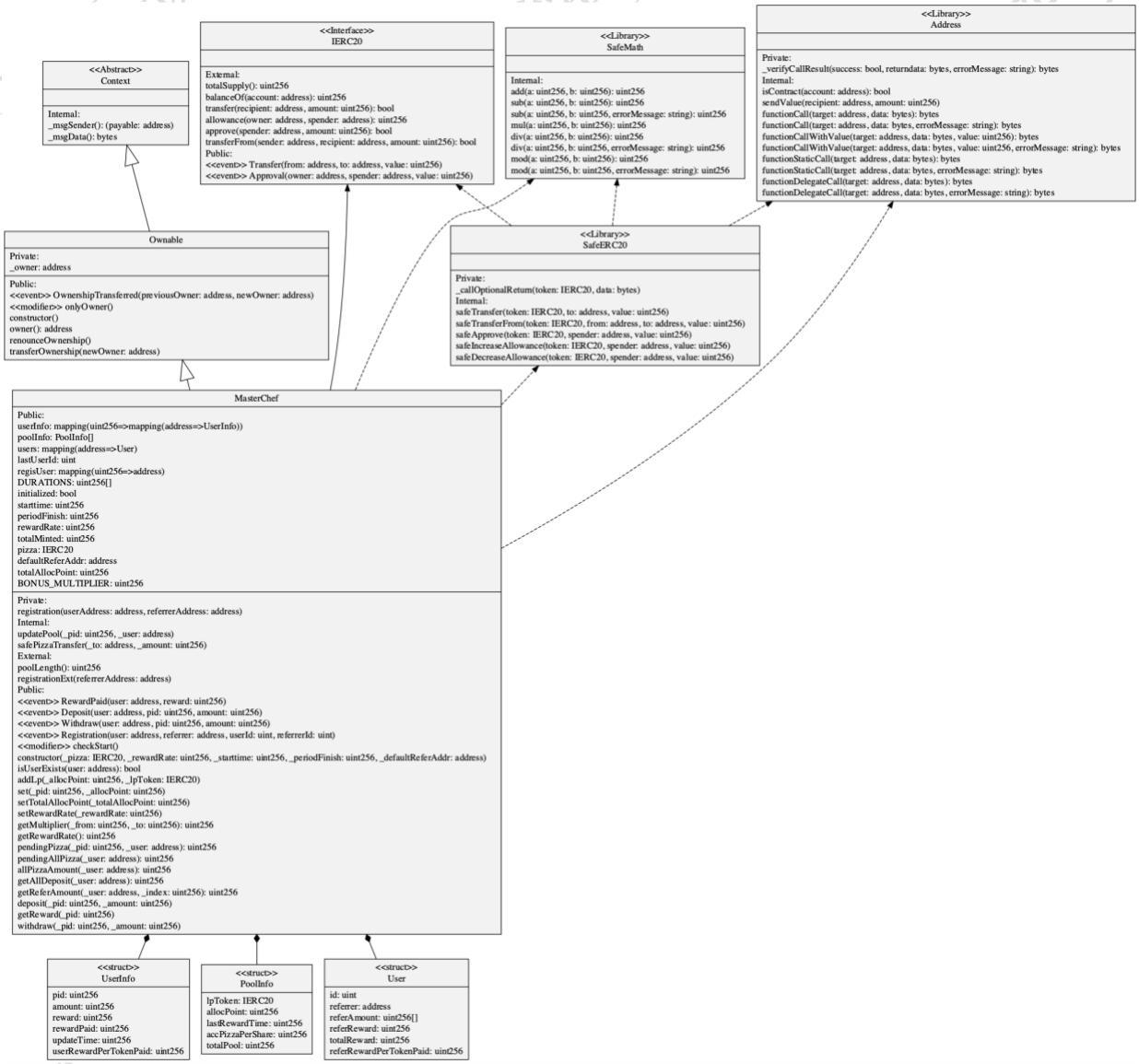
4 Summary of Security Audit

The contract project is a liquidity mining project, which obtains Pizza income by collateralizing LP; the quality of the contract code is relatively poor, and it does not have the function of extracting LP in an emergency. There are many key variables in the contract used to calculate the users mining revenue are controlled by the contract owner. Considering that the contract project has not yet deployed transactions, the overall security status of this audit is: **Warning status.**

Note: The Pizza Token contract is not in the scope of this audit.

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

5 Smart Contract UML



Appendix A. Explanation of Security Risk Status Levels

Security risk status statement	
1	**good status** 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** 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** 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** 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.