AUDIT

Jujube Finance

/WildernessSec/

Version 1.0.0
Serial No. 202300000032
Presented by WildernessSec
Feb 20, 2023

/WildernessSec/

Contents

1 - Introduction

This document includes the results of the audit performed by the WildernessSec team on the Jujube Finance project, at the request of the Jujube Finance Team. This assessment was conducted between February 12th and February 20nd, 2023

Project Account In aptos testnet address:

https://explorer.aptoslabs.com
account/0x77777772a393c3958ba62ca4b1ce4a4da139a3a7f43313837d61f4c3ef
b2c711

Audited Contract Files:

The calculated SHA-256 values for the files are as follows:

```
router.move
ec7d5efee49eada17a5f59cd430b4ab1a2ee8009ad7d97eb602428d821cc

a3c1 liquidity_pool.move
fa0e1962e759b6a50dfb270f11ee9d19e36c4fe621b1d24540fa98d0cdc22

d83 trade_fee.move
70a9d285986af13dc4d5cb0aaeecc6186258059774fb3a5dc136ab7d37f9

dffb trade_pool.move
ff629af4a74d78a052185201e51167695e16d8703ed815eff1b9a6c446bce
```

```
97c dao_storage.move
3826360e07a101de1f852985be5ae4f98edb69476350071240a0165fbdaa

6518 dao_fee.move
a5210644b91774f30ad9e0f87593a3859784d6f270329de037d73f05bd5f

446b trade_fee.move
70a9d285986af13dc4d5cb0aaeecc6186258059774fb3a5dc136ab7d37f9

dffb jujube_coin.move
ff9791d6520ade8ec6c29e71547c2a9f3b9ec008c61c34b836e6cf2da1174

302 jujube_chef.move
b15ad81400a5d638266dc9cda6dba036e44a7e2335d67aa76c35e30166e

7384b jujubeLock.move
0b1abc9af5764c189155647b15aed4968fd9e45df2f27e395e8715507a781

183 kingdom.move
b18e13eacd36445a2d065b2fa0dab0ecb56475e8c0503a0dc3a2c665b3de

36dd kingfee.move
e0986491e3016a9d1a79b43f670193ffe45ca410288bad897b3224340238

b6f5 kingchef.move
1a1e196c127cd1e49e8c58918ac29888475fa017c8c12f81c0d40dd8458a2

172 kingwar.move
0dd7504396cc00b5c983629a6bf19a7d86f01ea96f3f03baca48d752ca391

420 jujubeIDO.move
71c23b70b7a5e07f39b6c024f39c39c752ab94ab1268c4ca2fd2bb8d4049

3780 merkle.move
8c68a61dd8cea7de9f5b69ef7676893d6aa8df3c1a9cfbeedd3b0caad5d77
```

```
20e jujube_helper.move
de260a9cf81bc4aa342b9848401d450b6498706ff8750f374c95f7baca3d1

aeb jujube_nft_chef.move
e93f249aa254b083a77d6ba533b74b48a09331336595f35a4e46bfed6cd2
6b35
```

The jujube team asked us to review the move code for swap,kingdom and chef. Because of the amount of code, our team spent a week to finish the audit report.

Our general overview of the Jujube team's code is that it is very well structured and designed. Code unit test coverage is high, most functions contain tests, and key functions contain multiple unit tests.

2 - About WildernessSec

WildernessSec aims to provide a secure and reliable solution for auditing blockchain smart contracts. The project's main goal is to protect user assets and ensure the correctness of contract execution, preventing smart contracts from being attacked or experiencing other abnormal situations due to security issues.

To achieve this, WildernessSec offers a range of technologies and tools, including contract static analysis, contract dynamic analysis, code review, and vulnerability fixing. These technologies and tools are designed to help developers write more secure smart contracts, while ensuring that users' interests are not compromised.

Overall, the goal of WildernessSec is to provide higher security and stability to the blockchain ecosystem, promoting the healthier and sustainable development of blockchain technology.

3 - Introduction to Jujube Finance

JUJUBE is a decentralized finance (DeFi) platform that offers several features, including a swap function, a chef function, a trade pool function, and a kingdom function.

JUJUBE offers a range of features that are designed to provide users with a comprehensive and user-friendly DeFi experience. Whether users are looking to swap tokens, earn rewards through farming, or participate in platform governance, JUJUBE provides a platform that can meet their needs.

4 - Severity level reference

Every issue in this report was assigned a severity level from the following:

Critical severity issues need to be fixed as soon as possible.

High severity issues will probably bring problems and should be fixed.

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better

fixed at some point in the future.

5 - List of issues by severity

A. Critical

- N/A

B.High

- WS_JUJUBE01 **(Function Permissions Are Not Set Correctly)**

  Description：
  jujube_helper::admin_batch_settle() This function does not set administrator privileges, so anyone can settle the fee, although the caller cannot get the fee, but the platform will not be able to settle the fee regularly.

  Recommendation:
  Adding administrator privileges

```
public entry fun admin_batch_settle<X,Y,Z,W,T,P,N,U>(sender:&signer){
 if(type_info::type_name<X>() !=type_info::type_name<NoneCoin>()){
       kingdom::admin_settle_fee<X>(sender);
   };
 if(type_info::type_name<Y>() !=type_info::type_name<NoneCoin>()){
       kingdom::admin_settle_fee<Y>(sender);
   };
   kingdom::init_king_history_and_withdraw_time(sender);
}
```

- WS_JUJUBE02 **(User's Nft Record Is Not Cleared)**

Description：
jujube_nft_chef::emergency_withdraw() This method does not clear user.nfts, resulting in subsequent users having the nft in their data, but no deposit power to receive the reward.

Recommendation:
Adding administrator privileges

```
public entry fun emergency_withdraw<CoinType>(
        account: &signer
    ) acquires MasterChefData, UserInfo, Events, PoolInfo {
        /***more code
        user_info.amount = 0;
        user_info.reward_debt = 0;
    user_info.nfts = vector::empty<String>();
        /***more code
    }
```

C. Medium

- WS_JUJUBE03**(Pseudo Random Numbers)**

  Description：
  kingdom::random_change_buff() This method has the problem of pseudo-random number when obtaining random number to change buff.

  Recommendation:
  Consider using random numbers generated by oracles instead of onchain data.

- WS_JUJUBE04**(Pseudo Random Numbers)**

  Description:
  kingdom::betting_national_fortunes() This method has the problem of pseudo-random number when obtaining random number to get national_fortunes.

  Recommendation:
  Consider using random numbers generated by oracles instead of onchain data.

D. Low

- WS_JUJUBE05**(Lack Of Ownership Transfer Logic)**

  Description：
  From our observation, the platform does not implement the admin permission transfer, the current contract will be determined when the admin deploys the contract, and all the functions on the platform are related to the admin account. If the private key is compromised, the platform becomes uncontrollable

  Recommendation:
  It is recommended that wherever admin permission is required, instead of using the address in mof. toml, you wrap a get_admin_address() function and add the set_admin_address() method to set it.

- WS_JUJUBE06**(Jujube Coin does not set the freeze function)**

  Description：
  We see that jujube_coin retains the freeze ability, but the freeze function is not set in jujube_coin.mv. The freeze ability is useful. When there are some errors in the system, or some users obtain jujube through abnormal means, the user token authority can be frozen through the freeze function

  Recommendation:

```
public entry fun freeze(
    account: &signer,
    freeze_account: address,
) {
    coin::freeze_coin_store(freeze_account, &freeze_cap);
}
```

Thank you for choose

/WildernessSec/