CRACKEN

Smart Contract Security Audit

Project: Kung Fu Panda

Sep 12, 2022

Contract Address

0xcdBc993e04Ac3397Db3740194366d75fD3bf3937

Table of Contents

Disclaimer

The contents of this report reflect only the CRACKEN TECH audit team's understanding of the current progress and status of the security of the code audited, to verify the integrity of the code provided for the scope of this audit. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Given the size of the project, the findings detailed here are not to be considered exhaustive, and further testing and audit are recommended after the issues covered are fixed. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report.

The review does not address the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. If the audited source files are smart contract files, risks or issues introduced by using data feeds from off-chain sources are not extended by this review either.

Audit Review

The source code of the Kung Fu Panda was audited in order to acquire a clear impression of how the project was implemented. The Cracken Tech audit team conducted in-depth research, analysis, and scrutiny, resulting in a series of observations. A detailed list of each issue found, and vulnerabilities in the source code will be included in the audit report. The problems and potential solutions are given in this report, we will identify common sources for such problems and comments for improvement.

The auditing process will follow a routine as special considerations by Cracken:

- Review of the specifications, sources, and instructions provided to Cracken to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Cracken describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
- Symbolic execution is analyzing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

Project Review

Token Summary

Parameter	Result
Token Name	Kung Fu Panda
Token Symbol	KFP
Token Decimal	6
Total Supply	419,000,000
Platform	BSC
Buy Tax Fee	10%
Sell Tax Fee	10%
Contract Creation Date	Sep 12, 2022
Liquidity Status	Not available when audit
Liquidity Lockup Time	365 days after pool ends
Compiler Version	v0.8.14+commit.80d49f37
Optimization	No with 200 runs
Contract Address	0xcdBc993e04Ac3397Db3740194366d75fD3bf3937
Deployer Address	0x0525d51e8ae7da563cd81bb5c0c7cab276cc2f17
Owner Address	0x0525d51e8ae7da563cd81bb5c0c7cab276cc2f17

Source Code

CRACKEN was commissioned by Kung Fu Panda to perform an audit based on the following smart contract:

https://bscscan.com/address/0xcdBc993e04Ac3397Db3740194366d75fD3bf3937

Smart Contract Vulnerability Checks

Vulnerability	Auto-Scan	Manual-Scan	Result
Unencrypted Private Data On-Chain	Complete	Complete	Low / No Risk
Code With No Effects	Complete	Complete	Low / No Risk
Message call with hardcoded gas amount	Complete	Complete	Low / No Risk
Hash Collisions with Multiple Variable Length Arguments	Complete	Complete	Low / No Risk
Unexpected Ether balance	Complete	Complete	Low / No Risk
Presence of unused variables	Complete	Complete	Low / No Risk
Right-To-Left-Override control character (U+202E)	Complete	Complete	Low / No Risk
Typographical Error	Complete	Complete	Low / No Risk
DoS With Block Gas Limit	Complete	Complete	Low / No Risk
Arbitrary Jump with Function Type Variable	Complete	Complete	Low / No Risk
Insufficient Gas Grieving	Complete	Complete	Low / No Risk
Incorrect Inheritance Order	Complete	Complete	Low / No Risk
Write to Arbitrary Storage Location	Complete	Complete	Low / No Risk
Requirement Violation	Complete	Complete	Low / No Risk
Missing Protection against Signature Replay Attacks	Complete	Complete	Low / No Risk
Weak Sources of Randomness from Chain Attributes	Complete	Complete	Low / No Risk
Authorization through tx. origin	Complete	Complete	Low / No Risk
Delegate call to Untrusted Callee	Complete	Complete	Low / No Risk

Vulnerability	Auto-Scan	Manual-Scan	Result
Use of Deprecated Solidity Functions	Complete	Complete	Low / No Risk
Assert Violation	Complete	Complete	Low / No Risk
Reentrancy	Complete	Complete	Low / No Risk
Unprotected SELF-DESTRUCT Instruction	Complete	Complete	Low / No Risk
Unprotected Ether Withdrawal	Complete	Complete	Low / No Risk
Outdated Compiler Version	Complete	Complete	Low / No Risk
Integer Overflow and Underflow	Complete	Complete	Low / No Risk
Function Default Visibility	Complete	Complete	Low / No Risk

Manual Code Review

Classification of Issues

Severity	Description
High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
Informational	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
High-Risk	1
Medium-Risk	0
Low-Risk	0
Informational	0
Total	1

🔴 **High-Risk: functions make cause the rug or scam project. Must be fixed.**

Set max buy / sell tax fee

Description:

The owner can change the buy & sell fees up to 100%

[HIGH-RISK]

```
function updateLpFee(uint256 newLpFee) public onlyOwner {
        lpFee = newLpFee;
    }

    function updateNodeFee(uint256 modeFee) public onlyOwner {
        _taxNode = modeFee;
    }

    function updateTaxFoundation(uint256 newTaxFoundation) public onlyOwner {
        _taxFoundation = newTaxFoundation;
    }

    function updateTaxBurn(uint256 newTaxBurn) public onlyOwner {
        _taxBurn = newTaxBurn;
    }
```

Recommendation:

We recommend adding a requirement to limit the max fee amount.

Privileged Functions

onlyOwner

Function Name	Parameters	Visibility
addIdolist	address user,uint256 idonum	Public
approve	address account, uint256 amount	Public
burn	address account, uint256 amount	Internal
decreaseAllowance	address spender, uint256 subtractedValue	Public
increaseAllowance	address spender, uint256 addedValue	Public
ownerDeposit	uint256 amount	Public
ownerWithdrew	uint256 amount	Public
renounceOwnership	None	Public
setController	address controllerAddr	Public
transfer	address recipient, uint256 amount	External
transferFrom	address sender,address recipient,uint256 amount	Public
transferOwnership	address newOwner	Public
updateEndTime	uint256 newtime	Public
updateExecutors	address executor, bool status	Public
updateFoundationAddress	address newAddress	Public
updateKfpPool	address newpool	Public
updateLpAddress	address setAdress	Public
updateLpFee	uint256 newLpFee	Public

updateMinAmount	uint256 newMin	Public
updateNodeAddress	address newAddress	Public
updateNodeFee	uint256 modeFee	Public
updateReceiveExclude	address receiver, bool isExclude	Public
updateSendExclude	address sender, bool isExclude	Public
updateTaxBurn	uint256 newTaxBurn	Public
update TaxFoundation	uint256 newTaxFoundation	Public

Contract Ownership

The contract ownership of Kung Fu Panda is not currently being renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0x0525d51e8ae7da563cd81bb5c0c7cab276cc2f17 which can be viewed: HERE

The owner wallet has the power to call the functions displayed on the privileged functions list above, if the owner wallet is compromised these privileges could be exploited.

We recommend the team renounce ownership at the right timing if possible, or gradually migrate to a time lock with governing functionalities in respect of transparency and safety considerations.

Liquidity Overview

Liquidity Information

Parameter	Result
Pair Address	Not available
KFP Reserves	0.00 KFP
BNB Reserves	0.00 BNB
Liquidity Value	$0.00 USDT
Liquidity Ownership	The token does not have liquidity at the moment of the audit

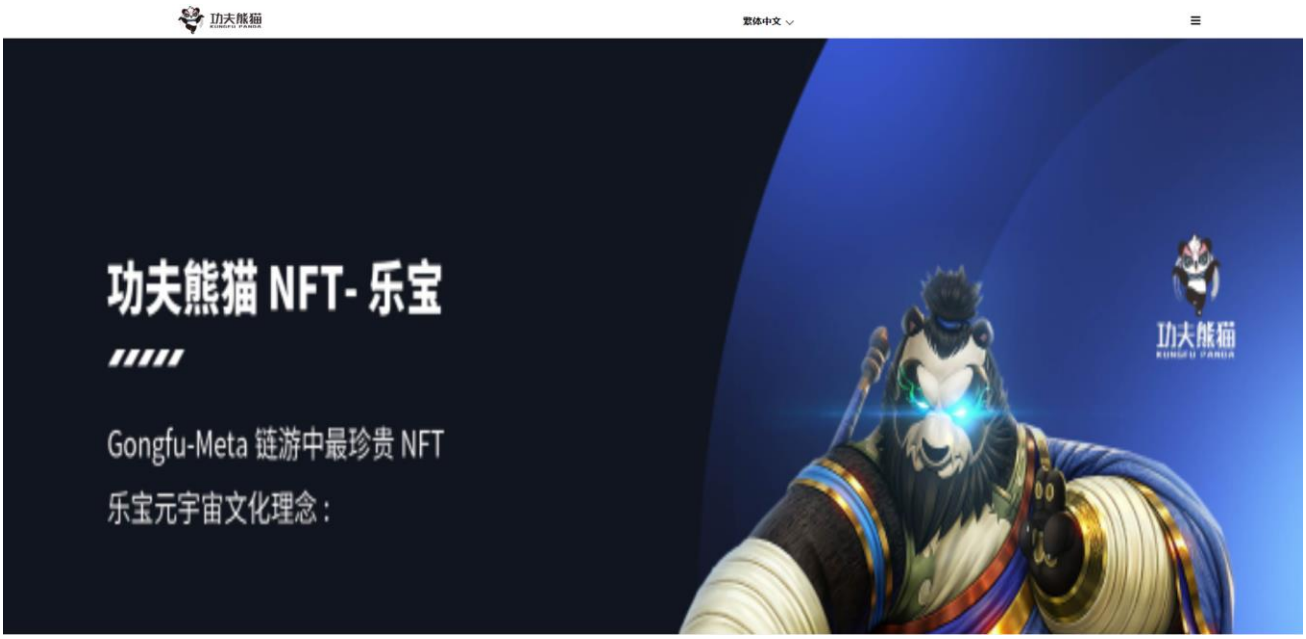
Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0x0525d51e8ae7da563cd81bb5c0c7cab276cc2f17	419,000,000	100.0000%

Social Media Check

Social Media Type	Link	Result
Website	https://dapp.kungfupanda.site/	Checked
Twitter	https://twitter.com/KungFuPanda_66/	Checked
Telegram	https://t.me/KungFuPanDaKFP/	Checked

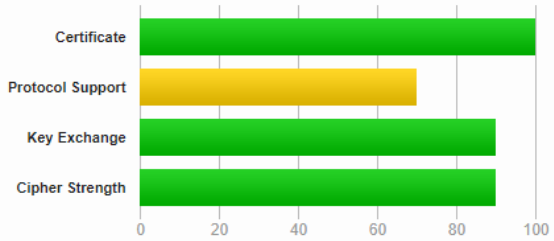
Website Review



Summary

Overall Rating

B



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server supports TLS 1.0 and TLS 1.1. Grade capped to B. MORE INFO »

This site works only in browsers with SNI support.

This server supports TLS 1.3.

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. MORE INFO »

Certificate #1: EC 256 bits (SHA384withECDSA)

Server Key and Certificate #1

Subject	*.kungfupanda.site Fingerprint SHA256: fc32e9aa0ee0fd465f9932b2f60457b346bdf0dc90180ac108bb4bcf1b363e96 Pin SHA256: sVMoqQTBqBjAK8jWA/yEj4yAGX+IxoDrP1v9avUJYhc=
Common names	*.kungfupanda.site
Alternative names	*.kungfupanda.site kungfupanda.site
Serial Number	042d141c6869b9cdbab4529aac4531423a8a
Valid from	Tue, 16 Aug 2022 01:26:23 UTC
Valid until	Mon, 14 Nov 2022 01:26:22 UTC (expires in 2 months and 1 day)
Key	EC 256 bits
Weak key (Debian)	No
Issuer	E1 AIA: http://e1.i.lencr.org/
Signature algorithm	SHA384withECDSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://e1.o.lencr.org
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

- PC is not friendly
- Contains no code errors
- SSL is secured
- Few spelling errors are founded

Audit Conclusion

- The owner cannot pause trading.
- The owner cannot mint new tokens.
- The owner cannot add blacklist users.
- The owner cannot set the max transaction amount.
- **The owner can change the buy/sell fee up to 100%.**

  (All functions cannot be used if the ownership is renounced)