



Smart Contract Security Audit

Project: TomCat Token

Sep 30, 2022



Contract Address

0x3b368B649994DC33E08315E400F2ECd23723974f

Table of Contents

- 1 Disclaimer**
- 2 Audit Review**
- 3 Project Review**
- 4 Smart Contract Vulnerability Checks**
- 5 Manual Code Review**
- 6 Owner Privileges**
 - 6.1 Contract Ownership
 - 6.2 Liquidity Overview
- 7 Tokenomics**
- 8 Social Media Check**
- 9 Website Review**
- 10 Audit Conclusion**

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 Tom Cat 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	TomCat
Token Symbol	TCat
Token Decimal	9
Total Supply	1,000,000,000,000,000
Platform	BSC
Buy Tax Fee	9%
Sell Tax Fee	9%
Contract Creation Date	Sep 28, 2022
Liquidity Status	Not Available
Liquidity Lockup Time	Not Available
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	No with 200 runs
Contract Address	0x3b368B649994DC33E08315E400F2ECd23723974f
Deployer Address	0x2faa634c329c2cfc93bc269f8dbd017e23fb44b1
Owner Address	0x2faa634c329c2cfc93bc269f8dbd017e23fb44b1

Source Code

CRACKEN was commissioned by Tom Cat to perform an audit based on the following smart contract:

<https://bscscan.com/address/0x3b368B649994DC33E08315E400F2ECd23723974f>

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	0
 Medium-Risk	1
 Low-Risk	0
 Informational	1
Total	2

● **Medium-Risk:** functions make cause a few bugs of the project. Should be fixed.

Set max buy / sell tax fee

Description:

The owner can change the buy & sell fees up to 16%.

```
function setFee(uint256 redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy,
uint256 taxFeeOnSell) public onlyOwner {

    require(redisFeeOnBuy < 5, "Redis cannot be more than 5.");

    require(redisFeeOnSell < 5, "Redis cannot be more than 5.");

    require(taxFeeOnBuy < 11, "Tax cannot be more than 11.");

    require(taxFeeOnSell < 11, "Tax cannot be more than 11.");

    _redisFeeOnBuy = redisFeeOnBuy;

    _redisFeeOnSell = redisFeeOnSell;

    _taxFeeOnBuy = taxFeeOnBuy;

    _taxFeeOnSell = taxFeeOnSell;

}
```

● Informational: Implementation of certain corrective actions or accepting the risk.

Set max buy / sell tax fee

Description:

The owner can add blacklist users.

[MEDIUM-RISK]

```
function blacklistAddress(address account, bool value) public onlyOwner{

    _isBlacklisted[account] = value;

    emit BlacklistAddress(account, value);

}

event BlacklistMultiAddresses(address[] accounts, bool value);

function blacklistMultiAddresses(address[] calldata accounts, bool value) public
onlyOwner{

    for(uint256 i = 0; i < accounts.length; i++) {

        _isBlacklisted[accounts[i]] = value;

    }

    emit BlacklistMultiAddresses(accounts, value);

}
```

Privileged Functions

onlyOwner

Function Name	Parameters	Visibility
approve	address spender, uint256 amount	Public
blacklistAddress		Public
lacklistMultiAddresses	address spender, uint256 subtractedValue	Public
excludeFromFees	address spender, uint256 addedValue	Public
excludeMultipleAccountsFromFees	address[] calldata accounts, bool excluded	Public
renounceOwnership	None	Public
rescueForeignTokens	address _tokenAddr, address _to, uint _amount	Public
setFee	uint256 redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy, uint256 taxFeeOnSell	Public
setNewAppAddress	address payable appaddr	Public
setNewBurnAddress	address payable burnaddr	Public
setNewBuybackAddress	address payable buybackaddr	Public
setNewMarketingAddress	address payable markt	Public
setPresaleContract	address payable wallet	External
setSnipeBlocks	uint8 _blocks	External
transfer	address recipient, uint256 amount	Public
transferFrom	address sender, address recipient, uint256 amount	Public
transferOwnership	address newOwner	Public

Contract Ownership

The contract ownership of Tom Cat is not currently 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 `0x2faa634c329c2cfc93bc269f8dbd017e23fb44b1` 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	<code>0xf111f4b9b2f3befba83a3955cc391fabe4b5e1a4</code>
TCat Reserves	0.00 TCat
BNB Reserves	0.00 BNB
Liquidity Value	\$0 USD
Liquidity Ownership	The token does not have liquidity at the moment of the audit

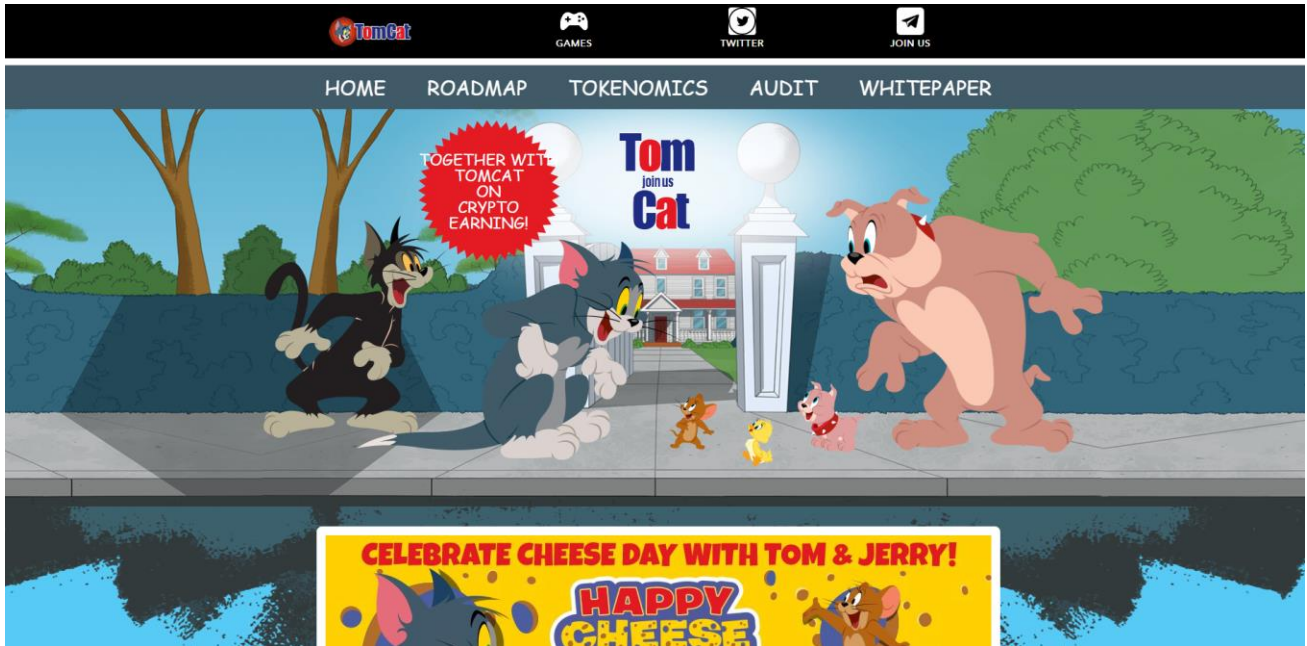
Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0x2faa634c329c2cfc93bc269f8dbd017e23fb44b1	1,000,000,000,000,000	100.0000%

Social Media Check

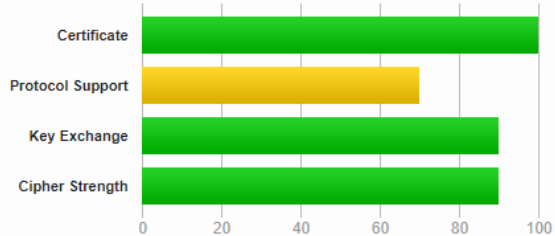
Social Media Type	Link	Result
Website	https://tomcatbsc.link/	Checked
Twitter	https://twitter.com/TomCatBinance/	Checked
Telegram	https://t.me/TomCatGlobal/	Checked

Website Review



Summary

Overall Rating



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 »](#)

Certificate #1: RSA 2048 bits (SHA256withRSA)



Server Key and Certificate #1



Subject	tomcatbsc.link Fingerprint SHA256: f2bc6d21bc14769eef3bc97d1e3330b4bdf1555a0ee3ab3470de7d32e94c28c3 Pin SHA256: xDE+5Vg5U/Xr+I7OaQPMRC2y257CwjK4f3ZMLpBSPs=
Common names	tomcatbsc.link
Alternative names	*.tomcatbsc.link tomcatbsc.link
Serial Number	042a1c74fc813623a7f4b29202be065f16fd
Valid from	Wed, 28 Sep 2022 17:30:09 UTC
Valid until	Tue, 27 Dec 2022 17:30:08 UTC (expires in 2 months and 27 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	R3 AIA: http://r3.i.lencr.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://r3.o.lencr.org
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

- Mobile Friendly
- Contains no code errors
- SSL is Secured
- No spelling errors

Audit Conclusion

- The owner cannot pause trading
- The owner cannot mint new tokens
- **The owner can blacklist users [Medium-Risk]**
- The owner cannot change the max tx amount
- The owner can change buy/sell fees up to 16%.
(The fees cannot be changed if the owner renounced the ownership)

AUDIT IS PASSED