CFG NINJA AUDITS

Security Assessment

Tombili Token

August 4, 2023

Audit Status: Pass

Audit Edition: Pinksale

POWERED BY

BLADE POOL

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
🔴 Critical	Danger or Potential Problems.
🟠 High	Be Careful or Fail test.
🟢 Low	Pass, Not-Detected or Safe Item.
ℹ Informational	Function Detected

Manual Code Review Risk Results

Contract Priviledge	Description
🟢 Buy Tax	5%
🟢 Sale Tax	5%
🟢 Cannot Sale	Pass
🟢 Cannot Sale	Pass
🟢 Max Tax	25%
🟢 Modify Tax	Yes
🟢 Fee Check	Pass
🟢 Is Honeypot?	Not Detected
🟢 Trading Cooldown	Not Detected
🟢 Can Pause Trade?	Pass

Contract Priviledge	Description
🟢 Pause Transfer?	Not Detected
🟢 Max Tx?	Pass
🟢 Is Anti Whale?	Not Detected
🟢 Is Anti Bot?	Not Detected
🟢 Is Blacklist?	Not Detected
🟢 Blacklist Check	Pass
🟢 is Whitelist?	Not Detected
🟢 Can Mint?	Pass
🟢 Is Proxy?	Not Detected
🟢 Can Take Ownership?	Not Detected
🟢 Hidden Owner?	Not Detected
ℹ Owner	0xecf34cb9d03b8777f919fa9a123a39f88bebf280
🟢 Self Destruct?	Not Detected
🟢 External Call?	Not Detected
🟢 Other?	Not Detected
🟢 Holders	1
🟢 Auditor Confidence	Low

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview

Token Summary

Parameter	Result
Address	0x65F7A222BaF435721Cb28613f5aA0C6174Efd9a0
Name	Tombili
Token Tracker	Tombili (TOMB)
Decimals	9
Supply	100,000
Platform	Binance Smart Chain
compiler	v0.8.19+commit.7dd6d404
Contract Name	Tombili
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/token/0x65F7A222BaF435721Cb28613f5aA0C6174Efd9a0#code
Payment Tx	Corporate

Main Contract Assessed
Contract Name

Name	Contract	Live
Tombili	0x65F7A222BaF435721Cb28613f5aA0C6174Efd9a0	Yes

TestNet Contract Assessed
Contract Name

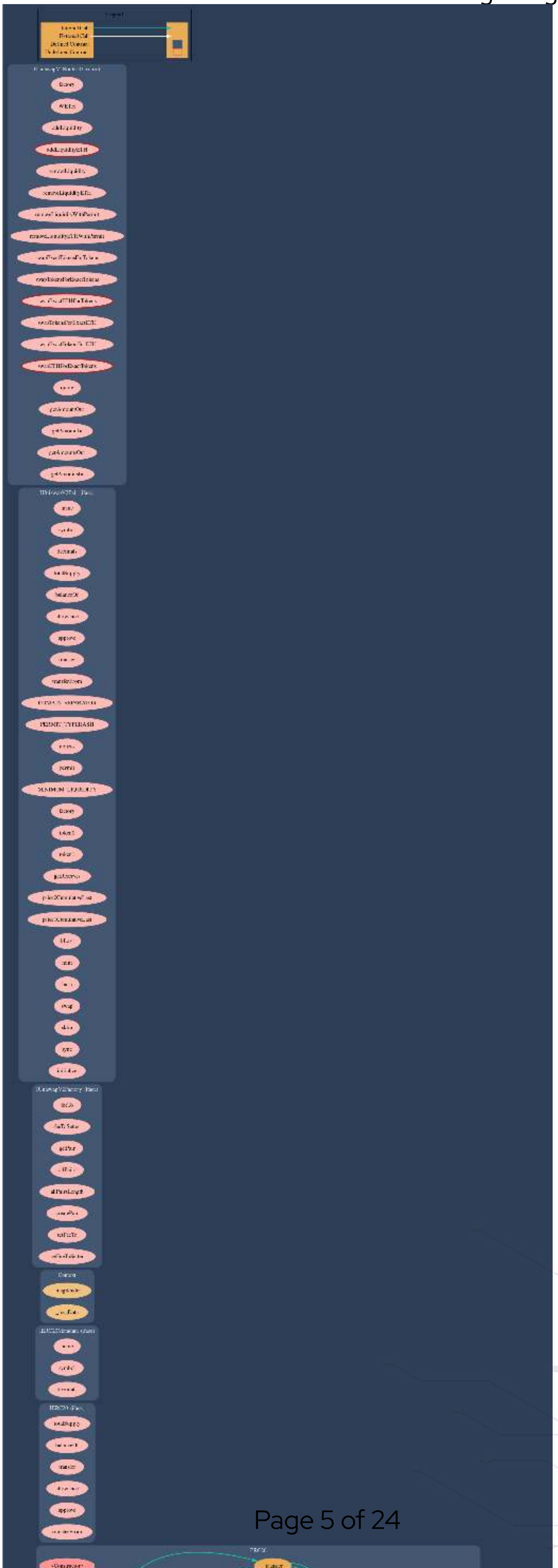
Name	Contract	Live
Tombili	0xcE69bFE070865c503F0E538f5b8eb95aC02830bf	Yes

Solidity Code Provided

SolID	File Sha-1	FileName
TOMB	d37056bc2e1deb6e90946514858bf5d434cec35c	Token.sol
TOMB	eed9e370a5a0bb0141a486f06aad38dc002a98d6	ERC20.sol
TOMB	922a6c259db4815b1fea14ad3432b4e60ff8c156	IERC20.sol
TOMB	efdbd0c99914615ee965f77afaf26b322714f9f7	IERC20Metadata.sol
TOMB	719844505df30bda93516e78eab1ced3bfe9ff4a	Context.sol
TOMB	63cc34195a232988d4d83db2aba05af1ae179fdd	ERC20Burnable.sol

Call Graph

The contract for Tombili has the following call graph structure.



Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	Token.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	Token.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	Token.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	Token.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	Token.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	Token.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	Token.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	Token.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	Token.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	Token.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	Token.sol	L: 0 C: 0

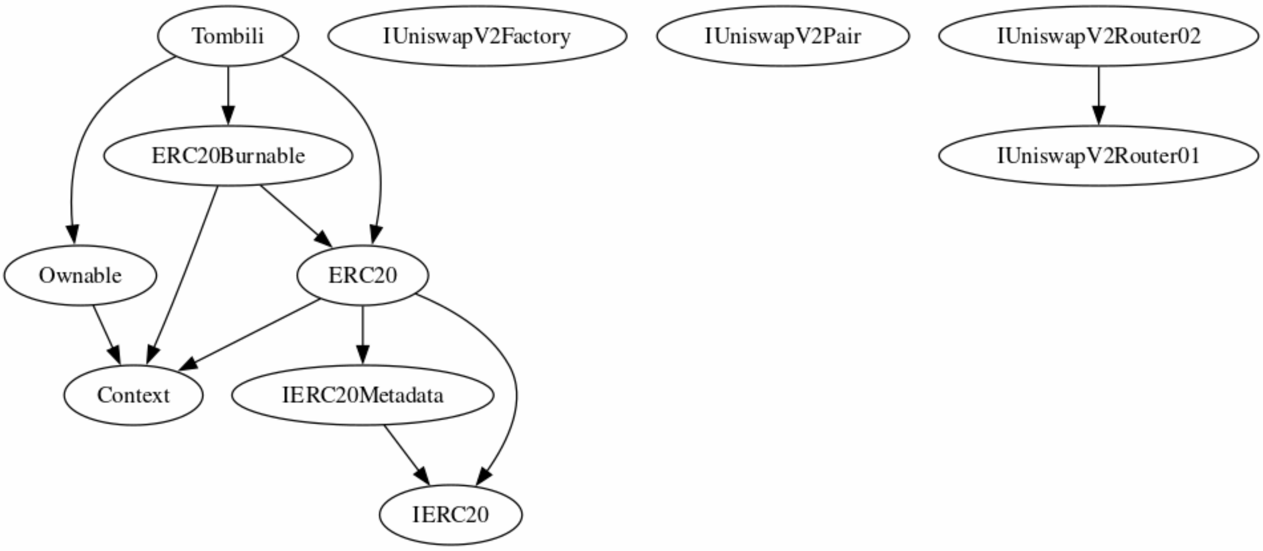
ID	Severity	Name	File	location
SWC-111	Pass	Use of Deprecated Solidity Functions.	Token.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	Token.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	Token.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	Token.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	Token.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	Token.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	Token.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	Token.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	Token.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	Token.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	Token.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	Token.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	Token.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	Token.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	Token.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-126	Pass	Insufficient Gas Griefing.	Token.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	Token.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	Token.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	Token.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	Token.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	Token.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	Token.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	Token.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	Token.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	Token.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	Token.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Inheritance

The contract for Tombili has the following inheritance structure.



Smart Contract Advance Checks

ID	Severity	Name	Result	Status
TOMB-01	Low	Potential Sandwich Attacks.	Pass	Not-Found
TOMB-02	Informational	Function Visibility Optimization	Fail	Detected
TOMB-03	Low	Lack of Input Validation.	Fail	Detected
TOMB-04	High	Centralized Risk In addLiquidity.	Fail	Detected
TOMB-05	Low	Missing Event Emission.	Pass	Not Detected
TOMB-06	Low	Conformance with Solidity Naming Conventions.	Pass	Not-Found
TOMB-07	Low	State Variables could be Declared Constant.	Pass	Not-Found
TOMB-08	Low	Dead Code Elimination.	Pass	Not-Found
TOMB-09	High	Third Party Dependencies.	Pass	Not Detected
TOMB-10	High	Initial Token Distribution.	Pass	Not-Found
TOMB-11	High	A function require success during a swapAndLiquify event.	Fail	Not Detected
TOMB-12	High	Centralization Risks In The X Role	Pass	Not-Found
TOMB-13	Informational	Extra Gas Cost For User..	Pass	Not Detected
TOMB-14	Medium	Unnecessary Use Of SafeMath	Pass	Not Detected

ID	Severity	Name	Result	Status
TOMB-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected
TOMB-16	Medium	Taxes can be up to 100%	Pass	Not Detected
TOMB-17	Logical Issue	Highly Permissive Role Access.,`	Pass	Not Detected
TOMB-18	Critical	Stop Transactions by using Enable Trade.	Pass	Not Detected

TOMB-02 | Function Visibility Optimization.

Category	Severity	Location	Status
Gas Optimization	Informational	Token.sol: L: 155 C: 14	Detected

Description

The following functions are declared as public and are not invoked in any of the contracts contained within the projects scope:

Function Name	Parameters	Visibility
updateSwapThreshold		public
setAMMPair		public
excludeFromFees		public
liquidityFeesSetup		public
lpTokensReceiverSetup		public
buybackFeesSetup		public
buybackAddressSetup		public
charityFeesSetup		public
charityAddressSetup		public
marketingFeesSetup		public
marketingAddressSetup		public
updateSwapThreshold		public

The functions that are never called internally within the contract should have external visibility

Remediation

We advise that the function's visibility specifiers are set to external, and the array-based arguments change their data location from memory to calldata, optimizing the gas cost of the function.

References:

external vs public best practices.

TOMB-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	🟢 Low	Token.sol: L: 155 C: 14	Detected

Description

The given input is missing the check for the non-zero address.

The given input is missing the check for the updateSwapThreshold,marketingAddressSetup, lpTokensReceiverSetup.

Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
    ...
     require(receiver != address(0), "Receiver is the zero address");
    ...
    ...
    require(value X limitation, "Your not able to do this function");
   ...
```

We also recommend customer to review the following function that is missing a required validation. updateSwapThreshold,marketingAddressSetup, lpTokensReceiverSetup.

TOMB-04 | Centralized Risk In addLiquidity.

Category	Severity	Location	Status
Coding Style	● High	Token.sol: L: 242 C: 14	Detected

Description

uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this), tokenAmount, 0, 0, owner(), block.timestamp);

The addLiquidity function calls the uniswapV2Router.addLiquidityETH function with the to address specified as owner() for acquiring the generated LP tokens from the TOMB-WBNB pool.

As a result, over time the _owner address will accumulate a significant portion of LP tokens.If the _owner is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Remediation

We advise the to address of the uniswapV2Router.addLiquidityETH function call to be replaced by the contract itself, i.e. address(this) , and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the _owner account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

1. Indicatively, here are some feasible solutions that would also mitigate the potential risk:
2. Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;
3. Assignment of privileged roles to multi-signature wallets to prevent single point of failure due to the private key;

Introduction of a DAO / governance / voting module to increase transparency and user involvement

Project Action

TOMB-11 | A function require success during a swapAndLiquify event..

Category	Severity	Location	Status
Optimization	🔴 High	Token.sol: L: 554 C: 14	Not Detected

Description

During the swap process there is a require function that will revert in case of failure, this can create a unstable situation for the contract. require(success, 'TaxesDefaultRouterWalletCoin: Fee transfer error');

Remediation

We recommend removing the success requirement during a swap event.

Project Action

Technical Findings Summary

Classification of Risk

Severity	Description
🔴 Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
🟠 High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
🟡 Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
🟢 Low	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
ℹ Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
🔴 Critical	1	0	0
🟠 High	1	0	0
🟡 Medium	0	0	0
🟢 Low	1	0	0
ℹ Informational	1	0	0
Total	4	0	0

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/tombili_token	Pass
Other	https://www.youtube.com/@TombiliToken	Pass
Website	https://tombili.xyz/	Pass
Telegram	https://t.me/TombiliTokenPortal	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Assessment Results

Score Results

Review	Score
Overall Score	93/100
Auditor Score	85/100
Review by Section	**Score**
Manual Scan Score	30/33
SWC Scan Score	37 /37
Advance Check Score	26 /30

The Following Score System Has been Added to this page to help understand the value of the audit, the maximun score is 100, however to attain that value the project most pass and provide all the data needed for the assessment. Our Passing Score has been changed to 80 Points, if a project does not attain 80% is an automatic failure. Read our notes and final assessment below.

Audit Passed

Assessment Results

Important Notes:

- No SWC or vulnerabilities were found.
- The code could use some minor improvements.
- the auto liquidity goes to an external wallet.
- Please DYOR on the project.

Auditor Score =85
Audit Passed

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

CFGNINJA has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

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. Information is provided 'as is, and CFGNINJA is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will CFGNINJA or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by CFGNINJA are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.

CFG NINJA AUDITS