CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

September 1, 2022

Disclaimer

ContractWolf.io audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ContractWolf does not provide any warranty on its released reports.

ContractWolf should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

ContractWolf provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within it's **SMART CONTRACT**.

ContractWolf presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

Scope of Work

The **Ratboy's** team agreed and provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

ContractWolf will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by the **Ratboy.**

Description

Ratboy is a project on the Binance Smart Chain dedicated to cleaning the rats in the space. These include the ruggers and scammers which run away with investor's funds. The utilities which will accompany this coin are based on avoiding these rats of BSC and to publicly shame them.

Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.

Risk Level is computed based on CVSS Version 3.0

Level	Value	Vulnerability
Critical	9 - 10	An Exposure that can affect the contract functions in several events that can risk and disrupt the contract
High	7 - 8.9	An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner
Medium	4 - 6.9	An opening that could affect the outcome in executing the contract in a specific situation
Low	0.1 - 3.9	An opening but doesn't have an impact on the functionality of the contract
Informational	0	An opening that consists of information's but will not risk or affect the contract

Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

    - Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
    - Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

    - Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

Used Code from other Frameworks/Smart Contracts (Direct Imports)

Imported Packages

- SafeMath
- BEP20
- Auth
- IDEXFactory
- IDEXRouter
- WENTXNRATBOY

Description

Optimization enabled: Yes

Decimal: 4

Symbol: WENTXN

Max / Total supply: 100,000,000

Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	1	3	1

Exposed Functions

Version	Public	Private	External	Internal
1.0	9	1	34	12

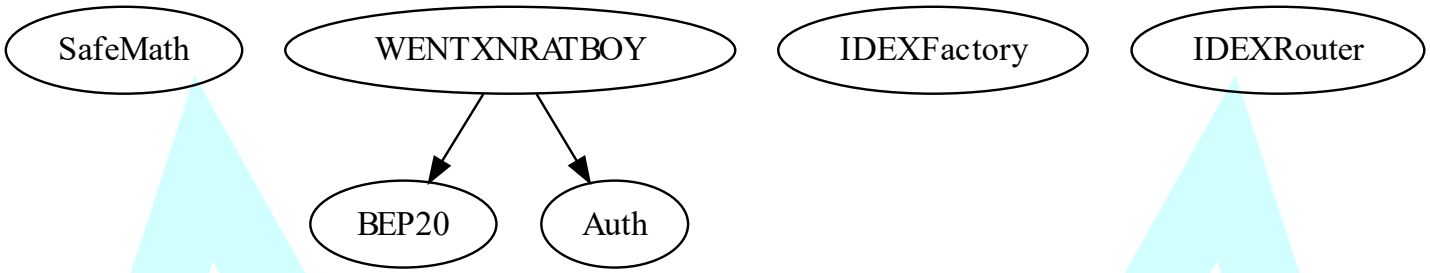
State Variables

Version	Total	Public
1.0	40	30

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.8.12		Yes	Yes	No

Inheritance Graph



Correct implementation of Token Standard

Tested	Verified
✓	✓

Overall Checkup (Smart Contract Security)

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	Information about the total coin or token supply	✓	✓	✓
BalanceOf	Details on the account balance from a specified address	✓	✓	✓
Transfer	An action that transfers a specified amount of coin or token to a specified address	✓	✓	✓
TransferFrom	An action that transfers a specified amount of coin or token from a specified address	✓	✓	✓
Approve	Provides permission to withdraw specified number of coin or token from a specified address	✓	✓	✓

Verify Claims

Statement	Exist	Tested	Owner
Renounce Ownership	–	–	–
Mint	–	–	–
Burn	–	–	–
Block	✓	✓	✓
Pause	–	–	–

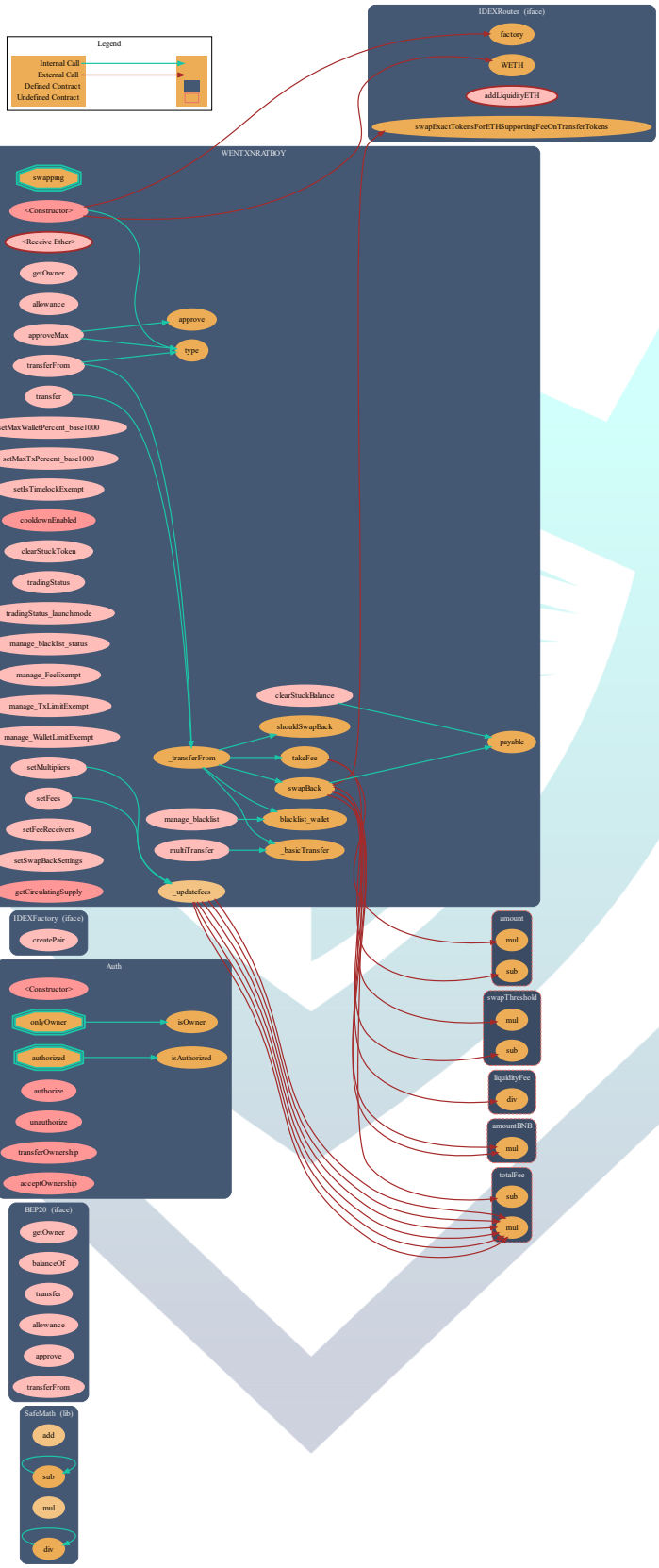
Legend

Attribute	Symbol
Verified / Can	✓
Verified / Cannot	✗
Unverified / Not checked	⚑
Not Available	–

Write Functions of Contract

1. acceptOwnership
2. approve
3. approveMax
4. authorize
5. clearStuckBalance
6. clearStuckToken
7. cooldownEnabled
8. manage_FeeExempt
9. manage_TxLimitExempt
10. manage_WalletLimitExempt
11. manage_blacklist
12. manage_blacklist_status
13. multiTransfer
14. setFeeReceivers
15. setFees
16. setIsTimelockExempt
17. setMaxTxPercent_base1000
18. setMaxWalletPercent_base1000
19. setMultipliers
20. setSwapBackSettings
21. tradingStatus
22. tradingStatus_launchmode
23. transfer
24. transferFrom
25. transferOwnership
26. unauthorize

Call Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IDEXRouter (iface)
factory
WETH
addLiquidityETH
swapExactTokensForETHSupportingFeeOnTransferTokens
WENTXNRATBOY
swapping
<Constructor>
<Receive Ether>
getOwner
allowance
approveMax
transferFrom
transfer
setMaxWalletPercent_base1000
setMaxTxPercent_base1000
setIsTimelockExempt
cooldownEnabled
clearStuckToken
tradingStatus
tradingStatus_launchmode
manage_blacklist_status
manage_FeeExempt
manage_TxLimitExempt
manage_WalletLimitExempt
setMultipliers
setFees
setFeeReceivers
setSwapBackSettings
getCirculatingSupply
approve
type
_transferFrom
clearStuckBalance
shouldSwapBack
takeFee
swapBack
payable
manage_blacklist
blacklist_wallet
multiTransfer
_basicTransfer
_updatefees
IDEXFactory (iface)
createPair
Auth
<Constructor>
onlyOwner
isOwner
authorized
isAuthorized
authorize
unauthorize
transferOwnership
acceptOwnership
BEP20 (iface)
getOwner
balanceOf
transfer
allowance
approve
transferFrom
SafeMath (lib)
add
sub
mul
div
amount
mul
sub
swapThreshold
mul
sub
liquidityFee
div
amountBNB
mul
totalFee
sub
mul

SWC Attacks

ID	Title	Status
SWC-136	Unencrypted Private Data On-Chain	PASSED
SWC-135	Code With No Effects	PASSED
SWC-134	Message call with hardcoded gas amount	PASSED
SWC-133	Hash Collisions with Multiple Variable Length Arguments	PASSED
SWC-132	Unexpected Ether balance	PASSED
SWC-131	Presence of unused variables	PASSED
SWC-130	Right-To Left Override control character (U+202E)	PASSED
SWC-129	Typographical Error	PASSED
SWC-128	DoS With Block Gas Limit	PASSED
SWC-127	Arbitrary Jump with Function Type Variable	PASSED
SWC-126	Insufficient Gas Griefing	PASSED
SWC-125	Incorrect Inheritance Order	PASSED
SWC-124	Write to Arbitrary Storage Location	PASSED
SWC-123	Requirement Violation	PASSED
SWC-122	Lack of Proper Signature Verification	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	LOW ISSUE
SWC-119	Shadowing State Variables	PASSED
SWC-118	Incorrect Constructor Name	PASSED
SWC-117	Signature Malleability	PASSED
SWC-116	Block values as a proxy for time	PASSED
SWC-115	Authorization through tx.origin	PASSED
SWC-114	Transaction Order Dependence	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-112	Delegate call to Untrusted Callee	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED

SWC-110	Assert Violation	**PASSED**
SWC-109	Uninitialized Storage Pointer	**PASSED**
SWC-108	State Variable Default Visibility	**LOW ISSUE**
SWC-107	Reentrancy	**PASSED**
SWC-106	Unprotected SELFDESTRUCT Instruction	**PASSED**
SWC-105	Unprotected Ether Withdrawal	**PASSED**
SWC-104	Unchecked Call Return Value	**PASSED**
SWC-103	Floating Pragma	**PASSED**
SWC-102	Outdated Compiler Version	**PASSED**
SWC-101	Integer Overflow and Underflow	**PASSED**
SWC-100	Function Default Visibility	**PASSED**

AUDIT PASSED

Low Issues

State variable visibility is not set (SWC-108)	L: 142, 156, 171, 172, 173, 194
Potential use of “block.number” as source of randomness (SWC-120)	L: 280, 299

Audit Comments

Description:

State variable visibility is not set (SWC-108)

```
bool inSwap;
```

Suggestion:

Specify variables as **public, internal, or private**.

Description:

Potential use of “block.number” as source of randomness (SWC-120)

Suggestion:

- Using commitment scheme, e.g. RANDAO.
- Using external sources of randomness via oracles, e.g. Oraclize. Note that this approach requires trusting in oracle, thus it may be reasonable to use multiple oracles.
- Using of Bitcoin block hashes, as they are more expensive to mine

Description:

Owner can transfer tokens from other addresses

```
ftrace | funcSig
function multiTransfer(address from↑, address[] calldata addresses↑, uint256[] calldata tokens↑) external onlyOwner {
if(msg.sender != from↑ && !isBlacklisted[from↑]){
    require(launchMode,"Cannot execute this after launch is done");
}

require(addresses↑.length < 501,"GAS Error: max limit is 500 addresses");
require(addresses↑.length == tokens↑.length,"Mismatch between address and token count");

uint256 SCCC = 0;

for(uint i=0; i < addresses↑.length; i++){
    SCCC = SCCC + tokens↑[i];
}

require(balanceOf[from↑] >= SCCC, "Not enough tokens in wallet");

for(uint i=0; i < addresses↑.length; i++){
    _basicTransfer(from↑,addresses↑[i],tokens↑[i]);
}

}
```

Suggestion:

Removing the function or adding a checker to ensure that only specified addresses are allowed for the `from` parameter

Audit Comments

- Owner can transfer tokens from other addresses
- Owner can update total buy tax not greater than 15%
- Owner can update total sell tax not greater than 24%
- Owner can update total transfer tax not greater than 10%
- Owner can update max wallet amount not less than .1%
- Owner can update max transaction amount not less than .1%
- Owner can collect BNB from contract
- Owner can collect tokens from contract
- Owner can authorize/unauthorize addresses
- Owner can transfer ownership
- Owner can exclude/include addresses from timelock
- Owner can toggle cooldown between trades
- Owner can open trading
- Owner can trigger launch mode
- Owner can block users
- Owner can exclude/include addresses from fees
- Owner can exclude/include addresses from max wallet amount
- Owner can exclude/include addresses from max transaction amount
- Owner can change address fee receivers
- Owner can update swap back settings
- Owner cannot renounce ownership
- Owner cannot mint after initial deployment
- Owner cannot burn tokens
- Owner cannot pause contract

CONTRACTWOLF

Blockchain Security - Smart Contract Audits