CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

August 10, 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 its **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

Mini Stars 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 **Mini Stars.**

Description

First ever ecosystem that brings you USDC yields, along with unprecedented access to elite NBA players, and insiders alike.

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

- Address
- Context
- IERC20
- IDEXPair
- IDEXRouter
- IDEXFactory
- Ownable
- IAntiSnipe
- Ministars

Description

Optimization enabled: Yes

Decimal: 9

Symbol: STARS

Max / Total supply: 1,000,000,000

Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	1	5	1

Exposed Functions

Version	Public	Private	External	Internal
1.0	12	1	45	16

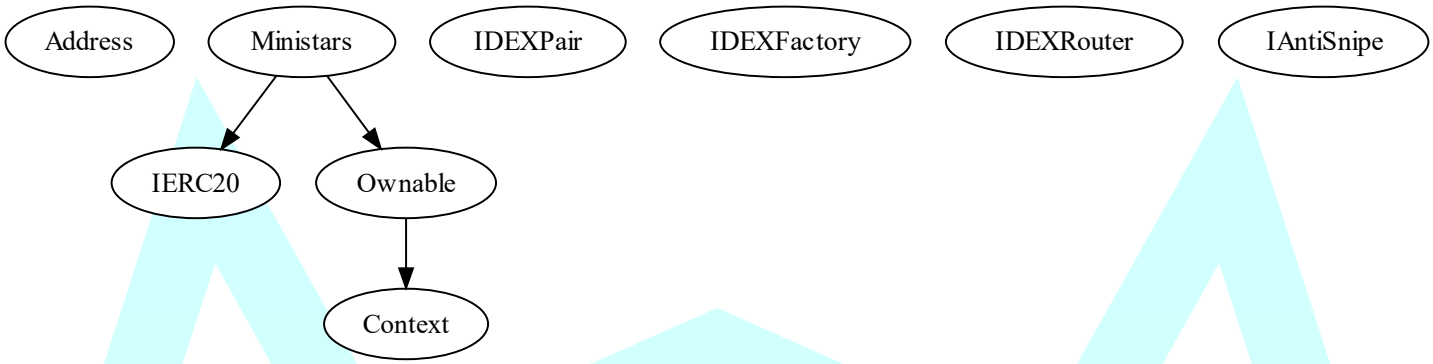
State Variables

Version	Total	Public
1.0	29	11

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.8.16		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	Deployer
Renounce Ownership	✓	✓	✓
Mint	–	–	–
Burn	–	–	–
Block	–	–	–
Pause	–	–	–

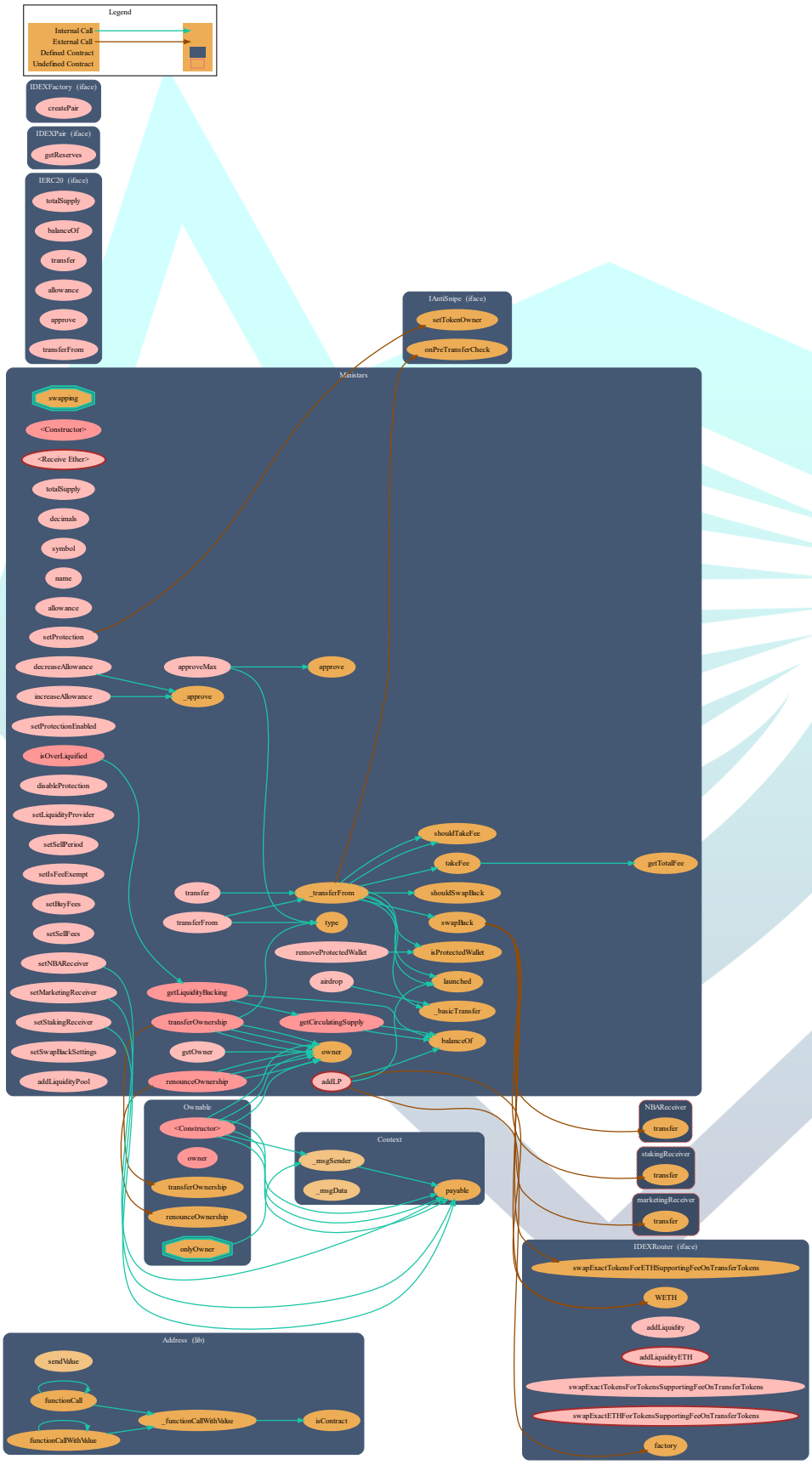
Legend

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

Write Functions of Contract (Testnet)

1. addLP
2. addLiquidityPool
3. airdrop
4. approve
5. approveMax
6. decreaseAllowance
7. disableProtection
8. increaseAllowance
9. removeProtectedWallet
10. renounceOwnership
11. setBuyFees
12. setIsFeeExempt
13. setLiquidityProvider
14. setMarketingReceiver
15. setNBAReceiver
16. setProtection
17. setProtectionEnabled
18. setSellFees
19. setSellPeriod
20. setStakingReceiver
21. setSwapBackSettings
22. transfer
23. transferFrom
24. transferOwnership

Call Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IDEXFactory (iface)
createPair
IDEXPair (iface)
getReserves
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
IAntiSnipe (iface)
setTokenOwner
onPreTransferCheck
Ministars
swapping
<Constructor>
<Receive Ether>
totalSupply
decimals
symbol
name
allowance
setProtection
decreaseAllowance
increaseAllowance
setProtectionEnabled
isOverLiquified
disableProtection
setLiquidityProvider
setSellPeriod
setIsFeeExempt
setBuyFees
setSellFees
setNBAReceiver
setMarketingReceiver
setStakingReceiver
setSwapBackSettings
addLiquidityPool
approveMax
approve
_approve
transfer
transferFrom
_transferFrom
type
shouldTakeFee
takeFee
getTotalFee
shouldSwapBack
swapBack
removeProtectedWallet
isProtectedWallet
airdrop
launched
getLiquidityBacking
transferOwnership
getCirculatingSupply
_basicTransfer
getOwner
owner
balanceOf
renounceOwnership
addLP
Ownable
<Constructor>
owner
transferOwnership
renounceOwnership
onlyOwner
Context
_msgSender
_msgData
payable
NBAReceiver
transfer
stakingReceiver
transfer
marketingReceiver
transfer
IDEXRouter (iface)
swapExactTokensForETHSupportingFeeOnTransferTokens
WETH
addLiquidity
addLiquidityETH
swapExactTokensForTokensSupportingFeeOnTransferTokens
swapExactETHForTokensSupportingFeeOnTransferTokens
factory
Address (lib)
sendValue
functionCall
_functionCallWithValue
isContract
functionCallWithValue

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**

THIS PROJECT WAS AUDITED VIA LOCAL FILE, AND IT'S NOT YET DEPLOYED ON LIVE NET

Low Issues

State variable visibility is not set (SWC-108)	L: 407, L: 408, L: 410, L: 411, L: 413, L: 414, L: 415, L: 416, L: 417, L: 418, L: 419, L: 420, L: 423, L: 424, L: 436, L: 437, L: 451
Potential use of "block.number" as source of randomness (SWC-120)	L: 572, L: 766

Audit Comments

- Deployer can set fees up to 10%
- Deployer can set sell tax up to 20% and sell period up to 7 days
- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can toggle protection status
- Deployer can set protection
- Deployer can disable protection
- Deployer can remove protected wallet address
- Deployer can set liquidity provider address
- Deployer can include/exclude addresses from fees
- Deployer can set NBA, marketing, and staking receiver addresses
- Deployer can set swap back settings
- Deployer can add LP address and toggle LP status
- Deployer can airdrop to multiple addresses
- Deployer cannot burn tokens
- Deployer cannot block user
- Deployer cannot pause contract
- Deployer cannot mint after initial deployment

CONTRACTWOLF

Blockchain Security - Smart Contract Audits