auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Wrapped Uni

DATED : 9 July 23'

MANUAL TESTING

Logical – Setting swap threshold to zero

Severity: High

function: setNumTokensBeforeSwap

Status: Not Resolved

Overview:

setting minimumTokensBeforeSwap to zero can disable transfer/sells due to contract trying to perfrom internal swap with 0 tokens.

```
function setNumTokensBeforeSwap(uint256 newLimit) external onlyOwner {
    minimumTokensBeforeSwap = newLimit;
}

if (overMinimumTokenBalance && !inSwapAndLiquify && !          isMarketPair[sender] &&
swapAndLiquifyEnabled) {
    if (swapAndLiquifyByLimitOnly) {
        contractTokenBalance = minimumTokensBeforeSwap;
    }
    swapAndLiquify(contractTokenBalance);
}
```

Suggestion

Ensure that minimumTokensBeforeSwap is always greater than one token.

```
function setNumTokensBeforeSwap(uint256 newLimit) external onlyOwner {
    minimumTokensBeforeSwap = newLimit;
    require(newLimit >= 1e18, "Can't set swap threshold to less than 1 token");
}
```

MANUAL TESTING

Logical – Malicious router

Severity: High

function: changeRouterVersion

Status: Not Resolved

Overview:

Router address can be updated to any arbitrary address. Updating router to a maliciuos contract can revert internal swaps which in result disables transfer/sells.

```
function changeRouterVersion(address newRouterAddress) public onlyOwner returns (address newPairAddress) {
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(newRouterAddress);

    newPairAddress = IUniswapV2Factory(_uniswapV2Router.factory()).getPair(address(this),
_uniswapV2Router.WETH());

    if (
        newPairAddress == address(0) //Create If Doesnt exist
    ) {
        newPairAddress =
            IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this), _uniswapV2Router.WETH());
    }

    uniswapPair = newPairAddress; //Set new pair address
    uniswapV2Router = _uniswapV2Router; //Set new router address

    isWalletLimitExempt[address(uniswapPair)] = true;
    isMarketPair[address(uniswapPair)] = true;
}
```

Suggestion

Ensure that router is immutable and can not be changed later, this can be achieved by removing **changeRouterVersion** function.

AUDIT SUMMARY

Project name – Wrapped Uni

Date: 9 July, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed With High Risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	2	0	0	1
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:
A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :
The code has undergone static analysis using Slither.

Testnet version:
The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:
https://testnet.bscscan.com/token/0x8CF4e073EF93a60D85EDD567a7fF285D39fB554b

Token Information

Token Name : Wrapped Uni

Token Symbol: WUNI

Decimals: 18

Token Supply: 1,000,000,000

Token Address:
0x580391D62614F6B3041d29893f2de23889914b58

Checksum:
5e3ba2795fd15d924231d690e5f3c135094594d4

Owner:
0x11170BA2FcDcA966cEa8970c6925c203086d5C3C
(at time of writing the audit)

Deployer:
0x11170BA2FcDcA966cEa8970c6925c203086d5C3C

TOKEN OVERVIEW

Fees:

Buy Fees: 1%

Sell Fees: 1%

Transfer Fees: 0%

Fees Privilege: owner

Ownership: owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: yes

Blacklist: No

Other Privileges: Initial distribution of the tokens

AUDIT METHODOLOGY

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

- Review of the specifications, sources, and instructions provided to Auditace 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 Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing 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.

VULNERABILITY CHECKLIST

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

CLASSIFICATION OF RISK

Severity	Description
Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
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.
Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
Critical	0
High-Risk	2
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	1

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	waiveOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	getTime	Public ❗		NO ❗
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO ❗

CONTRACT ASSESMENT

└	feeToSetter	External ❗		NO ❗
└	getPair	External ❗		NO ❗
└	allPairs	External ❗		NO ❗
└	allPairsLength	External ❗		NO ❗
└	createPair	External ❗	🛑	NO ❗
└	setFeeTo	External ❗	🛑	NO ❗
└	setFeeToSetter	External ❗	🛑	NO ❗
IUniswapV2Pair	Interface			
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	decimals	External ❗		NO ❗
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transfer	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
└	DOMAIN_SEPARATOR	External ❗		NO ❗
└	PERMIT_TYPEHASH	External ❗		NO ❗
└	nonces	External ❗		NO ❗
└	permit	External ❗	🛑	NO ❗
└	MINIMUM_LIQUIDITY	External ❗		NO ❗
└	factory	External ❗		NO ❗
└	token0	External ❗		NO ❗
└	token1	External ❗		NO ❗
└	getReserves	External ❗		NO ❗
└	price0CumulativeLast	External ❗		NO ❗
└	price1CumulativeLast	External ❗		NO ❗
└	kLast	External ❗		NO ❗
└	burn	External ❗	🛑	NO ❗
└	swap	External ❗	🛑	NO ❗
└	skim	External ❗	🛑	NO ❗
└	sync	External ❗	🛑	NO ❗
└	initialize	External ❗	🛑	NO ❗
IUniswapV2Router01	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	removeLiquidity	External ❗	🛑	NO ❗

CONTRACT ASSESMENT

└	removeLiquidityETH	External ❗	🛑	NO ❗
└	removeLiquidityWithPermit	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO ❗
└	swapExactTokensForTokens	External ❗	🛑	NO ❗
└	swapTokensForExactTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokens	External ❗	💵	NO ❗
└	swapTokensForExactETH	External ❗	🛑	NO ❗
└	swapExactTokensForETH	External ❗	🛑	NO ❗
└	swapETHForExactTokens	External ❗	💵	NO ❗
└	quote	External ❗		NO ❗
└	getAmountOut	External ❗		NO ❗
└	getAmountIn	External ❗		NO ❗
└	getAmountsOut	External ❗		NO ❗
└	getAmountsIn	External ❗		NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
WUNI	Implementation	Context, IERC20, Ownable		
└	<Constructor>	Public ❗	🛑	NO ❗
└	name	Public ❗		NO ❗
└	symbol	Public ❗		NO ❗
└	decimals	Public ❗		NO ❗
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	Public ❗		NO ❗
└	increaseAllowance	Public ❗	🛑	NO ❗
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	minimumTokensBeforeSwapAmount	Public ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗
└	_approve	Private 🔐	🛑	
└	setMarketPairStatus	Public ❗	🛑	onlyOwner
└	setIsTxLimitExempt	External ❗	🛑	onlyOwner
└	setIsExcludedFromFee	Public ❗	🛑	onlyOwner
└	enableDisableWalletLimit	External ❗	🛑	onlyOwner
└	setIsWalletLimitExempt	External ❗	🛑	onlyOwner
└	setNumTokensBeforeSwap	External ❗	🛑	onlyOwner
└	setMarketingWalletAddress	External ❗	🛑	onlyOwner

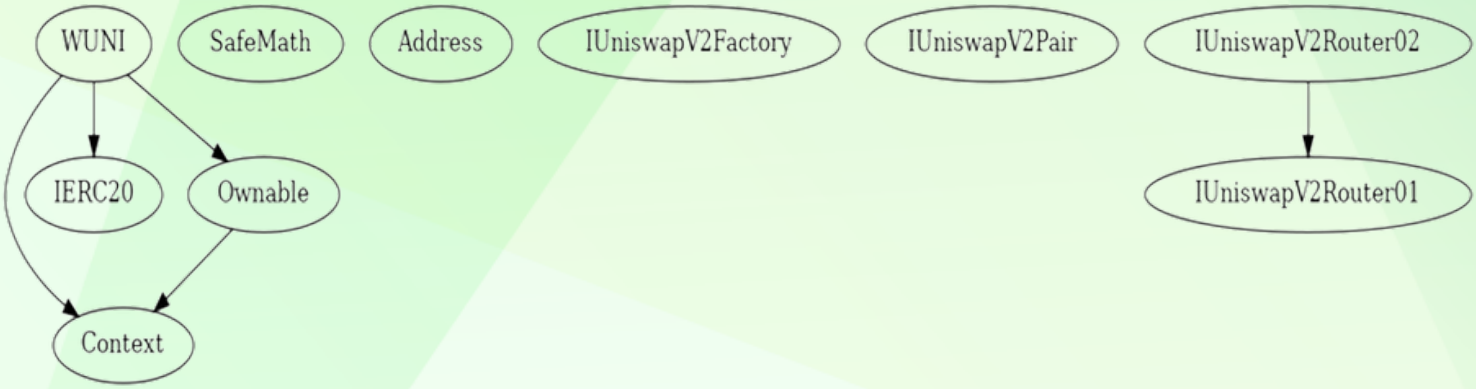
CONTRACT ASSESMENT

└	setSwapAndLiquifyEnabled	Public ❗	🛑	onlyOwner
└	setSwapAndLiquifyByLimitOnly	Public ❗	🛑	onlyOwner
└	getCirculatingSupply	Public ❗		NO ❗
└	transferToAddressETH	Private 🔐	🛑	
└	changeRouterVersion	Public ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗
└	transfer	Public ❗	🛑	NO ❗
└	transferFrom	Public ❗	🛑	NO ❗
└	_transfer	Private 🔐	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	takeFee	Internal 🔒	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

INHERITANCE TREE



POINTS TO NOTE

- **Owner is not able to change current fees (1% buy / 1% sell / 0% transfer)**
- Owner is not able to blacklist an address
- Owner is not able to disable buy/sell/transfers
- Owner is not able to set max wallet limit and minimum wallet limits
- Owner is not able to mint new tokens

STATIC ANALYSIS

```
Reentrancy in WUNI.transferFrom(address,address,uint256) (contracts/Token.sol#632-640):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#633)
                - recipient.transfer(amount) (contracts/Token.sol#602)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#633)
                - recipient.transfer(amount) (contracts/Token.sol#602)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (contracts/Token.sol#734-741)
        State variables written after the call(s):
        - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (contracts/Token.sol#634-638)
                - _allowances[owner][spender] = amount (contracts/Token.sol#552)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#553)
                - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (contracts/Token.sol#634-638)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#259) is too similar to IUniswapV2Router01.addLiquid
ity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#260)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

WUNI.slitherConstructorVariables() (contracts/Token.sol#416-760) uses literals with too many digits:
        - _totalSupply = 1000000000 * 10 ** _decimals (contracts/Token.sol#448)
WUNI.slitherConstructorVariables() (contracts/Token.sol#416-760) uses literals with too many digits:
        - minimumTokensBeforeSwap = _totalSupply.div(100000) (contracts/Token.sol#451)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

WUNI._buyLiquidityFee (contracts/Token.sol#435) should be constant
WUNI._buyMarketingFee (contracts/Token.sol#436) should be constant
WUNI._decimals (contracts/Token.sol#422) should be constant
WUNI._name (contracts/Token.sol#420) should be constant
WUNI._sellLiquidityFee (contracts/Token.sol#438) should be constant
WUNI._sellMarketingFee (contracts/Token.sol#439) should be constant
WUNI._symbol (contracts/Token.sol#421) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

WUNI._liquidityShare (contracts/Token.sol#441) should be immutable
WUNI._marketingShare (contracts/Token.sol#442) should be immutable
WUNI._maxTxAmount (contracts/Token.sol#449) should be immutable
WUNI._totalDistributionShares (contracts/Token.sol#446) should be immutable
WUNI._totalSupply (contracts/Token.sol#448) should be immutable
WUNI._totalTaxIfBuying (contracts/Token.sol#444) should be immutable
WUNI._totalTaxIfSelling (contracts/Token.sol#445) should be immutable
WUNI._walletMax (contracts/Token.sol#450) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

FUNCTIONAL TESTING

1- Adding liquidity (passed):

https://testnet.bscscan.com/tx/0x8d63d9895a170cdbee773808111dca967257698b3e60ec3cb2f91f8a954a69cb

2- Buying when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0x0e1805c2f975168103df8887ef2f9d7d47af63c35b03c98dd9dafec8f36aadc8

3- Selling when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0xaf71070dac52cf8382bb03ccf220c3078278b2b5fac67a22bcf1413dacebd364

4- Transferring when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0x604a476459f21448dfa0d4774027eba8b8f7d4e07e277748749ed461065f90ea

5- Buying(1% tax) (passed):

https://testnet.bscscan.com/tx/0xe62a9f5138534d3df135ba02eb6481cd5dd562c18210f70886678e029d2d1c19

6- Selling (1% tax) (passed):

https://testnet.bscscan.com/tx/0xde882d0f51d504de0813c1c1f6c09be53e2a207f74157ad9c91247aad76b90d7

FUNCTIONAL TESTING

4- Transferring (0% tax) (passed):

https://testnet.bscscan.com/tx/0x17c509fb06a4b736dd32574ccef7097a5f0043770976148a240719c0a2fb6dda

4- Internal swap (ETH sent to marketing wallet) (passed):

https://testnet.bscscan.com/address/0x11170ba2fcdca966cea8970c6925c203086d5c3c#internaltx

MANUAL TESTING

Logical – Setting swap threshold to zero

Severity: High

function: setNumTokensBeforeSwap

Status: Not Resolved

Overview:

setting minimumTokensBeforeSwap to zero can disable transfer/sells due to contract trying to perfrom internal swap with 0 tokens.

```
function setNumTokensBeforeSwap(uint256 newLimit) external onlyOwner {
    minimumTokensBeforeSwap = newLimit;
}

if (overMinimumTokenBalance && !inSwapAndLiquify && !          isMarketPair[sender] &&
swapAndLiquifyEnabled) {
    if (swapAndLiquifyByLimitOnly) {
        contractTokenBalance = minimumTokensBeforeSwap;
    }
    swapAndLiquify(contractTokenBalance);
}
```

Suggestion

Ensure that minimumTokensBeforeSwap is always greater than one token.

```
function setNumTokensBeforeSwap(uint256 newLimit) external onlyOwner {
    minimumTokensBeforeSwap = newLimit;
    require(newLimit >= 1e18, "Can't set swap threshold to less than 1 token");
}
```

MANUAL TESTING

Logical – Malicious router

Severity: High

function: changeRouterVersion

Status: Not Resolved

Overview:

Router address can be updated to any arbitrary address. Updating router to a maliciuos contract can revert internal swaps which in result disables transfer/sells.

```
function changeRouterVersion(address newRouterAddress) public onlyOwner returns (address newPairAddress) {
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(newRouterAddress);

    newPairAddress = IUniswapV2Factory(_uniswapV2Router.factory()).getPair(address(this),
_uniswapV2Router.WETH());

    if (
        newPairAddress == address(0) //Create If Doesnt exist
    ) {
        newPairAddress =
            IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this), _uniswapV2Router.WETH());
    }

    uniswapPair = newPairAddress; //Set new pair address
    uniswapV2Router = _uniswapV2Router; //Set new router address

    isWalletLimitExempt[address(uniswapPair)] = true;
    isMarketPair[address(uniswapPair)] = true;
}
```

Suggestion

Ensure that router is immutable and can not be changed later, this can be achieved by removing **changeRouterVersion** function.

MANUAL TESTING

Optimization – Redundant codes

Severity: Informational

function: ---

Status: Not Resolved

Overview:

Some features like max wallet / max transfer (buy/sell/transfer) / auto-liquidity are disabed by default and can not be enabled later. Its suggested to remove all the code sections related to this features in order to reduce overall gas usage of the contract

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace