Smart Contract Security Audit V1

NAVRAS Tech Token Smart Contract Audit

Nov 13, 2023

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Platform**: Ethereum
- **Contract Address**: Not deploy yet
- **Code Source**: https://goerli.etherscan.io/address/0x121dc6b583e77c61c2b49e270a1c1601e205968a#code

Contracts address deployed to test net (ETH)

NAVRAS Tech Token smart contracts on ETH test-net by the auditor to test every function.

https://goerli.etherscan.io/address/0x121dc6b583e77c61c2b49e270a1c1601e205968a

Token Information:

Name	NAVRAS Tech
Symbol	NAVRAS
Total supply	600,000,000
Decimals	18
Router	0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D
Buy Fee	6% Development fee = 3%, Operations fee = 3%
Sell Fee	6% Development fee = 3%, Operations fee = 3%
The initial owner of tokens	0xF8567e8161C885fb922Efdc819976f70f5F7D433
Development Address	0xD8EABD94e447e451c50E0094E0A89b9B552ae5a9
Operations Address	0x26639e869bc736DE38879Faffc0995984f82C54b
Swap Tokens at Amount	Collect 0.001% of total supply to swap to taxes

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well-Secured**.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 2 low, 0 very low-level issues and 1 note in all solidity files of the contract

The files:

NAVRAS.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
NAVRAS.sol	**2e7f88318b797e840239b05b5c89ebc7f124c6b7**	0x121DC6B583E77C61c2b49e270A1C1601e205968a

- Contract: NAVRAS
- Inherit: ERC20, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	**Passed**
symbol	✓	Read / public	**Passed**
decimals	✓	Read / public	**Passed**
totalSupply	✓	Read / public	**Passed**
allowance	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
decimals	✓	Read / public	**Passed**
BuyDevelopment	✓	Read / public	**Passed**
buyTaxes	✓	Read / public	**Passed**
BuyOperations	✓	Read / public	**Passed**
checkBlacklist	✓	Read / public	**Passed**
checkWhitelist	✓	Read / public	**Passed**

swapTokensAtAmount	✓	Read / public	**Passed**
isSwapping	✓	Read / public	**Passed**
DevelopmentAddress	✓	Read / public	**Passed**
OperationsAddress	✓	Read / public	**Passed**
startTradingBlock	✓	Read / public	**Passed**
pairAddress	✓	Read / public	**Passed**
SellDevelopment	✓	Read / public	**Passed**
SellOperations	✓	Read / public	**Passed**
sellTaxes	✓	Read / public	**Passed**
swapAndLiquifyEnabled	✓	Read / public	**Passed**
tradingEnabled	✓	Read / public	**Passed**
transferTaxes	✓	Read / public	**Passed**
uniswapRouter	✓	Read / public	**Passed**
disableTrading	✓	Write / public	**Passed**
enableTrading	✓	Write / public	**Passed**
transferFrom	✓	Write / public	**Passed**
transfer	✓	Write / public	**Passed**
decreaseAllowance	✓	Write / public	**Passed**
increaseAllowance	✓	Write /public	**Passed**
renounceOwnership	✓	Write / public	**Passed**
approve	✓	Write / public	**Passed**
setBuyTaxes	✓	Write / public	**Passed**
setSellTaxes	✓	Write / public	**Passed**
setDevelopmentAddress	✓	Write / public	**Passed**
setOperationsAddress	✓	Write / public	**Passed**
setBlacklist	✓	Write / public	**Passed**
setWhitelist	✓	Write / public	**Passed**
setSwapTokensAtAmount	✓	Write / public	**Passed**
toggleSwapping	✓	Write / public	**Passed**

withdrawStuckETH	✓	Write / public	**Passed**
withdrawStuckTokens	✓	Write / public	**Passed**
transferOwnership	✓	Write / public	**Passed**

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	**Passed**
2	Race conditions and Reentrancy. Cross-function race conditions.	**Passed**
3	Possible delays in data delivery.	**Passed**
4	Oracle calls.	**Passed**
5	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed with notes**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed with notes**
10	Methods execution permissions.	**Passed**
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	**Passed**
12	The impact of the exchange rate on the logic.	**Passed**
13	Private user data leaks.	**Passed**
14	Malicious Event log.	**Passed**
15	Scoping and Declarations.	**Passed**
16	Uninitialized storage pointers.	**Passed**
17	Arithmetic accuracy.	**Passed**

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can’t lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can’t have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can’t affect smart contract execution and can be ignored.

Audit Findings

Critical:

No Critical severity vulnerabilities were found.

High:

No High severity vulnerabilities were found.

Medium:

No Medium severity vulnerabilities were found.

Low:

#Use of block.timestamp for comparisons

Description
The value of block.timestamp can be manipulated by the miner. And conditions with strict equality is difficult to achieve - block.timestamp.

Remediation
Avoid use of block.timestamp

Status: Acknowledged.

#Owner privileges (In the period when the owner isn't renounced)

Description

The owner can add / remove any address to whitelist or blacklist.
The owner can change buy and sell fees in the contract.
The owner can enable / disable the trading.

```
function enableTrading() external onlyOwner {
        require(!tradingEnabled, "Trading is already enabled");
        tradingEnabled = true;
        startTradingBlock = block.number;
    }

    function disableTrading() external onlyOwner {
        require(tradingEnabled, "Trading is already disabled");
        tradingEnabled = false;
    }

    function setBuyTaxes(uint256 _newBuyDevelopment, uint256 _newBuyOperations)
external onlyOwner {
        BuyDevelopment = _newBuyDevelopment;
        BuyOperations = _newBuyOperations;
```

```
        buyTaxes = BuyDevelopment.add(BuyOperations);
        emit BuyFeesUpdated(BuyDevelopment, BuyOperations);
    }

    function setSellTaxes(uint256 _newSellDevelopment, uint256 _newSellOperations)
external onlyOwner {
        SellDevelopment = _newSellDevelopment;
        SellOperations = _newSellOperations;
        sellTaxes = SellDevelopment.add(SellOperations);
        emit SellFeesUpdated(SellDevelopment, SellOperations);
function setWhitelistStatus(address _wallet, bool _status) external onlyOwner {
        whitelisted[_wallet] = _status;
        emit Whitelist(_wallet, _status);
    }

    function setBlacklist(address _address, bool _isBlacklisted) external onlyOwner
{
        blacklisted[_address] = _isBlacklisted;
        emit Blacklist(_address, _isBlacklisted);
    }
```

Remediation

Make these functions internal in next version or the team should announce the investors before doing anything to give them time if they want to do anything.

P.S: This issue is common to the majority of those smart contracts.

Status: Acknowledged.

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Solidity compiler Bugs

Description

The smart contract uses 0.8.18 which isn't the latest one every upgrade solves low security issues; you can check the bugs from this link
https://goerli.etherscan.io/solcbuginfo

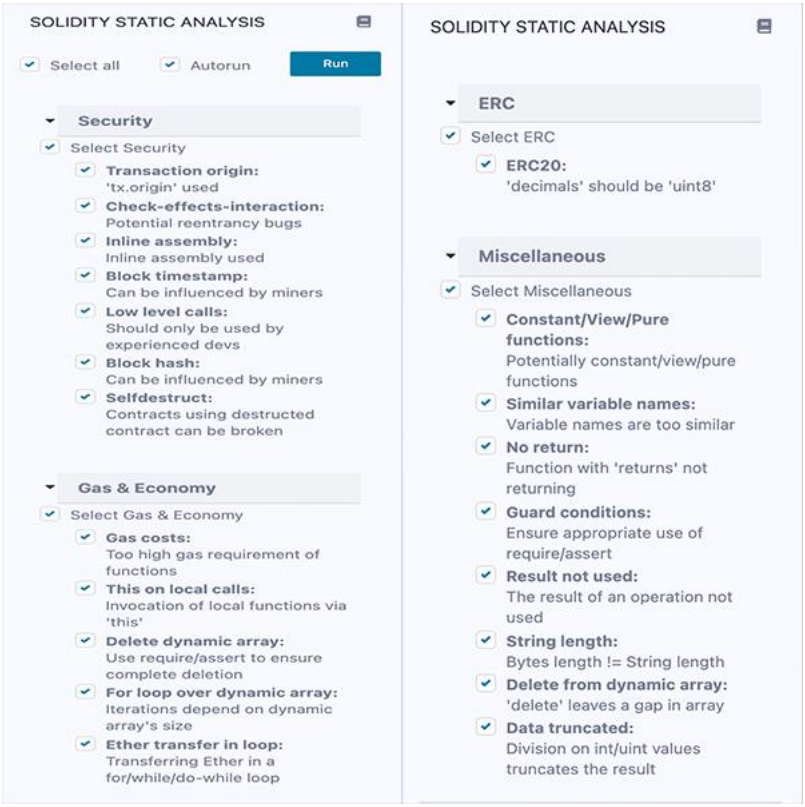
Remediation

Use 0.8.23 instead of 0.8.18

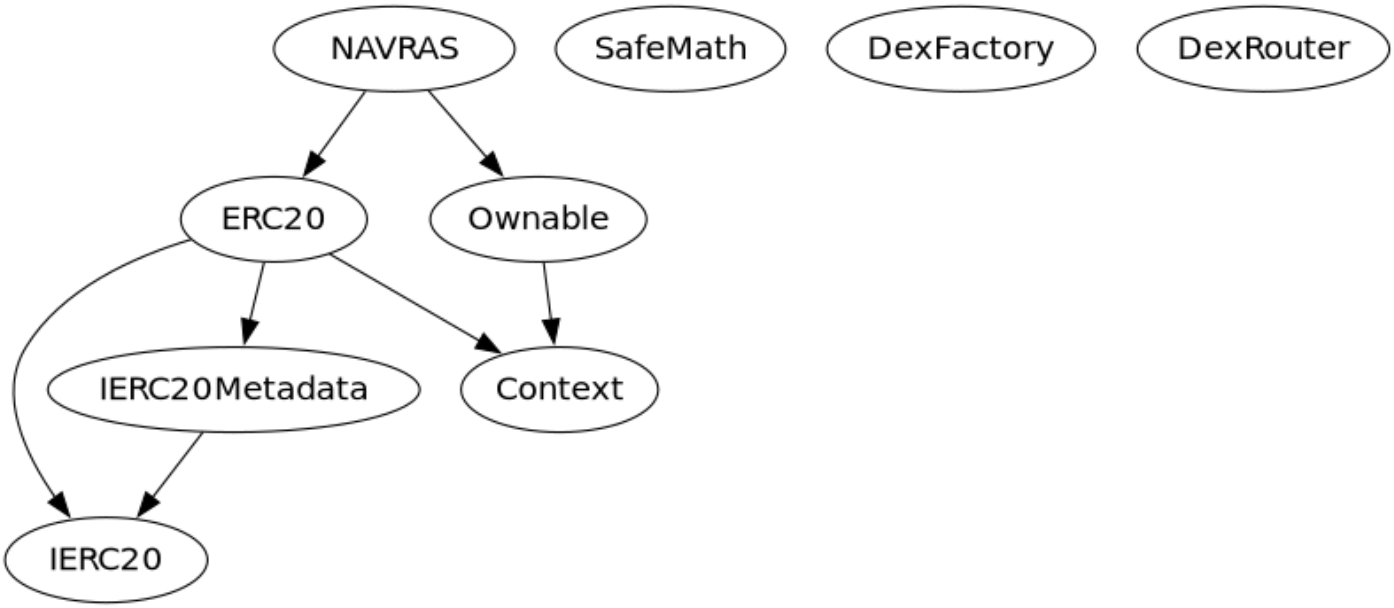
Status: Acknowledged.

Automatic Testing

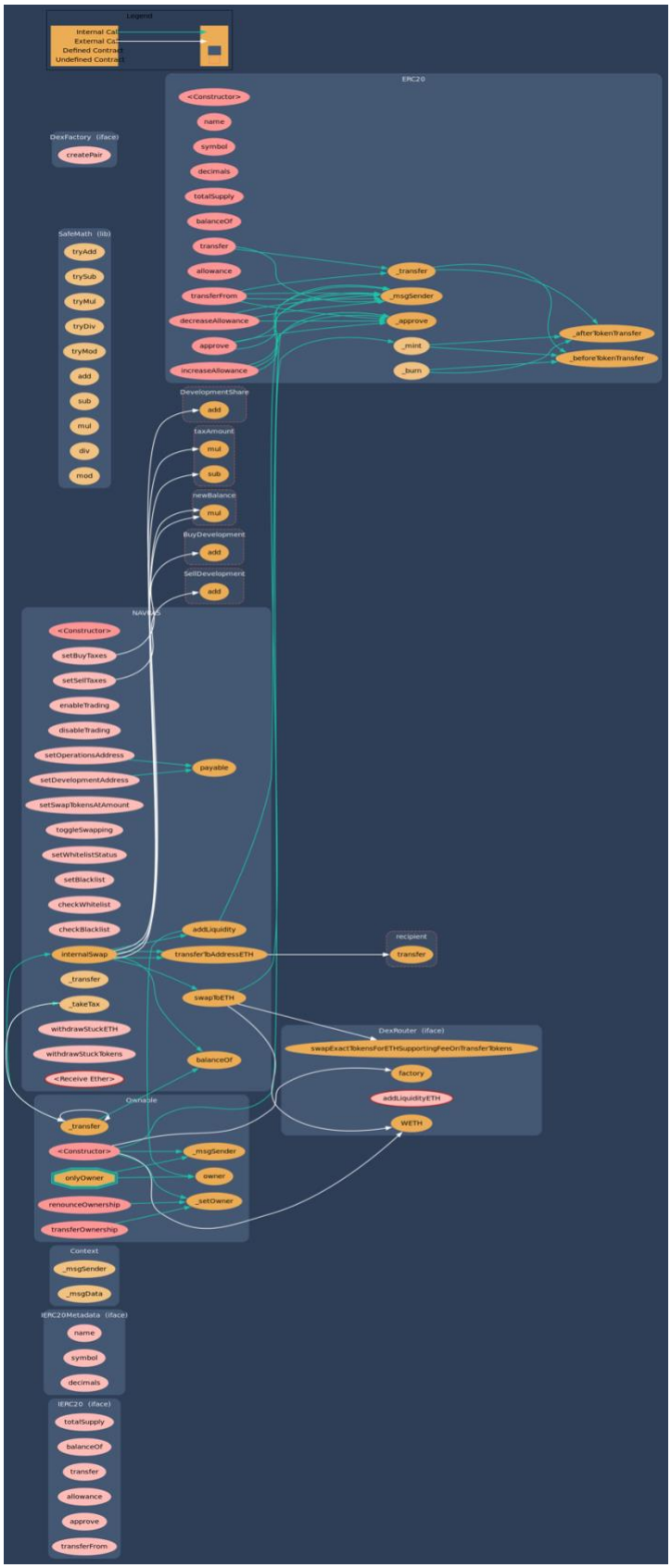
1- SOLIDITY STATIC ANALYSIS



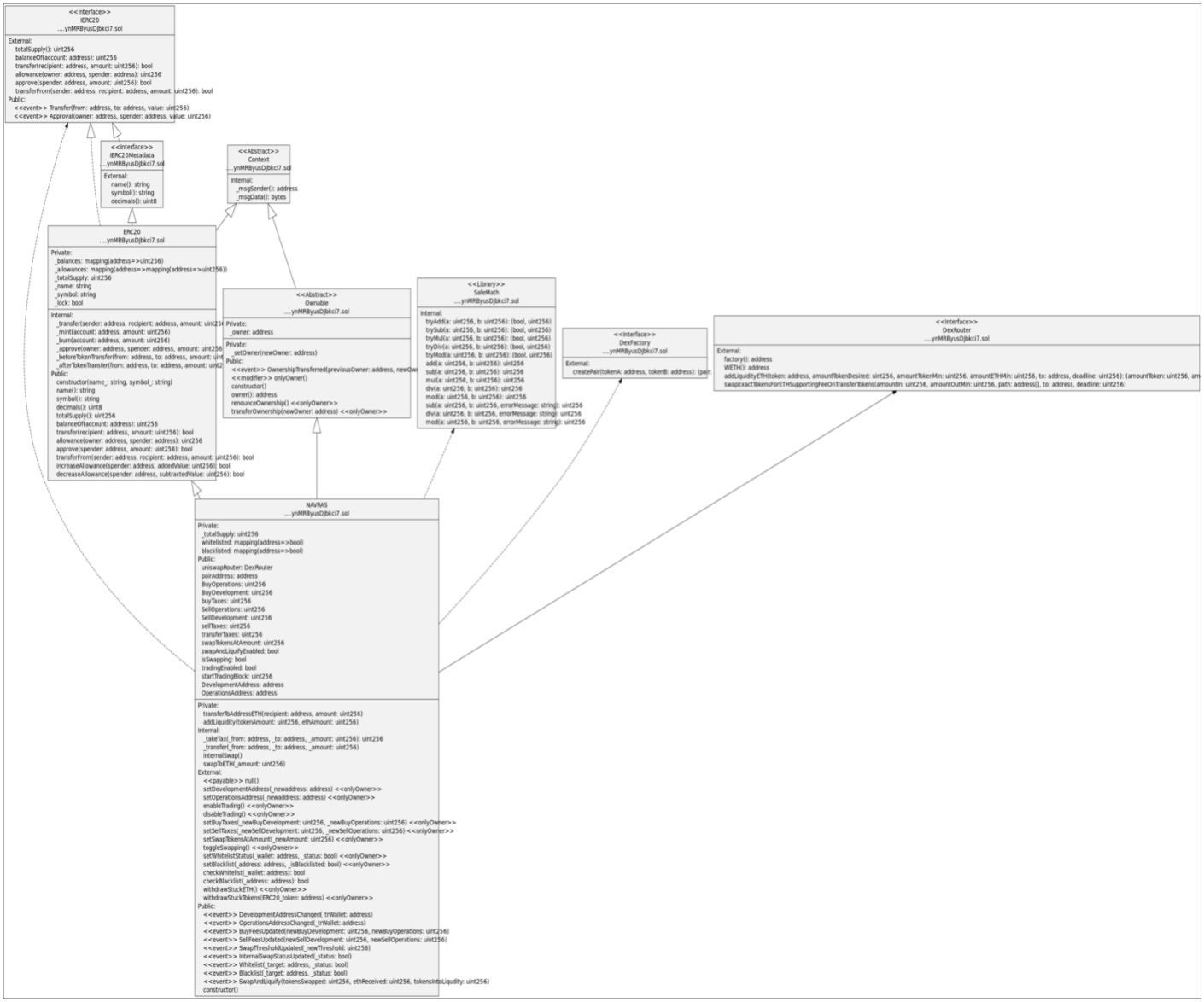
2- Inheritance graph



3- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transfer
allowance
transferFrom
decreaseAllowance
approve
increaseAllowance
_transfer
_msgSender
_approve
_mint
_burn
_afterTokenTransfer
_beforeTokenTransfer
DexFactory (iface)
createPair
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
DevelopmentShare
add
taxAmount
mul
sub
newBalance
mul
BuyDevelopment
add
SellDevelopment
add
NAVAX
<Constructor>
setBuyTaxes
setSellTaxes
enableTrading
disableTrading
setOperationsAddress
setDevelopmentAddress
setSwapTokensAtAmount
toggleSwapping
setWhitelistStatus
setBlacklist
checkWhitelist
checkBlacklist
internalSwap
_transfer
_takeTax
withdrawStuckETH
withdrawStuckTokens
<Receive Ether>
payable
addLiquidity
transferToAddressETH
swapToETH
balanceOf
recipient
transfer
DexRouter (iface)
swapExactTokensForETHSupportingFeeOnTransferTokens
factory
addLiquidityETH
WETH
Ownable
_transfer
<Constructor>
onlyOwner
renounceOwnership
transferOwnership
_msgSender
owner
_setOwner
Context
_msgSender
_msgData
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom

Unified Modeling Language (UML)



Functions signature

Function Name	Sighash	Function Signature
totalSupply	18160ddd	totalSupply()
balanceOf	70a08231	balanceOf(address)
transfer	a9059cbb	transfer(address,uint256)
allowance	dd62ed3e	allowance(address,address)
approve	095ea7b3	approve(address,uint256)
transferFrom	23b872dd	transferFrom(address,address,uint256)
name	06fdde03	name()
symbol	95d89b41	symbol()
decimals	313ce567	decimals()
name	06fdde03	name()
symbol	95d89b41	symbol()
decimals	313ce567	decimals()
totalSupply	18160ddd	totalSupply()
balanceOf	70a08231	balanceOf(address)
transfer	a9059cbb	transfer(address,uint256)
allowance	dd62ed3e	allowance(address,address)
approve	095ea7b3	approve(address,uint256)
transferFrom	23b872dd	transferFrom(address,address,uint256)
increaseAllowance	39509351	increaseAllowance(address,uint256)
decreaseAllowance	a457c2d7	decreaseAllowance(address,uint256)
owner	8da5cb5b	owner()
renounceOwnership	715018a6	renounceOwnership()
transferOwnership	f2fde38b	transferOwnership(address)
createPair	c9c65396	createPair(address,address)
factory	c45a0155	factory()
WETH	ad5c4648	WETH()
addLiquidityETH	f305d719	addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
swapExactTokensForETHSupportingFeeOnTransferTokens	791ac947	swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256)
setDevelopmentAddress	29b1c15c	setDevelopmentAddress(address)
setOperationsAddress	499b8394	setOperationsAddress(address)
enableTrading	8a8c523c	enableTrading()
disableTrading	17700f01	disableTrading()
setBuyTaxes	aa35822c	setBuyTaxes(uint256,uint256)
setSellTaxes	a11a1682	setSellTaxes(uint256,uint256)
setSwapTokensAtAmount	afa4f3b2	setSwapTokensAtAmount(uint256)
toggleSwapping	ef586f71	toggleSwapping()
setWhitelistStatus	0c424284	setWhitelistStatus(address,bool)
setBlacklist	153b0d1e	setBlacklist(address,bool)
checkWhitelist	1950c218	checkWhitelist(address)
checkBlacklist	e6807ca9	checkBlacklist(address)
withdrawStuckETH	f5648a4f	withdrawStuckETH()
withdrawStuckTokens	cb963728	withdrawStuckTokens(address)

Automatic general report

Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/NAVRASFinal.sol	2e7f88318b797e840239b05b5c89ebc7f124c6b7

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IERC20Metadata	Interface	IERC20		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	
Ownable	Implementation	Context		

└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
SafeMath	Library				
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
DexFactory	Interface				
└	createPair	External ❗	🛑	NO❗
DexRouter	Interface				
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
NAVRAS	Implementation	ERC20, Ownable			
└	<Constructor>	Public ❗	🛑	ERC20
└	setDevelopmentAddress	External ❗	🛑	onlyOwner
└	setOperationsAddress	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner
└	disableTrading	External ❗	🛑	onlyOwner
└	setBuyTaxes	External ❗	🛑	onlyOwner
└	setSellTaxes	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	toggleSwapping	External ❗	🛑	onlyOwner
└	setWhitelistStatus	External ❗	🛑	onlyOwner
└	setBlacklist	External ❗	🛑	onlyOwner
└	checkWhitelist	External ❗		NO❗
└	checkBlacklist	External ❗		NO❗
└	_takeTax	Internal 🔒	🛑	
└	_transfer	Internal 🔒	🛑	
└	internalSwap	Internal 🔒	🛑	
└	transferToAddressETH	Private 🔐	🛑	
└	swapToETH	Internal 🔒	🛑	
└	addLiquidity	Private 🔐	🛑	
└	withdrawStuckETH	External ❗	🛑	onlyOwner
└	withdrawStuckTokens	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “Well Secured”.

✓ No volatile code.
✓ No high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.