



Smart Contract Audit

FOR
Twitter X
DATED : 25 July 23'



MANUAL TESTING

Centralization – Excessive fees

Severity: **High**

function: updateTax

Status: Not Resolved

Overview:

Owner is able to put upto 50% tax on buy and sells separately.

```
function updateTax(uint256 _buyTax, uint256 _sellTax) external onlyOwner {  
    buyTax = _buyTax;  
    sellTax = _sellTax;  
    totalTax = _buyTax + _sellTax;  
    require(buyTax <= 49, "buy Fees cannot exceed 49%");  
    require(sellTax <= 49, "buy Fees cannot exceed 49%");  
    emit UpdateTax(buyTax, sellTax);  
}
```

Suggestion

Consider reducing the upper bound of maximum buy/sell tax. Usually 10% is suggested to be a reasonable upperbound limit for buy/sell/transfer tax.

```
function updateTax(uint256 _buyTax, uint256 _sellTax) external onlyOwner {  
    buyTax = _buyTax;  
    sellTax = _sellTax;  
    totalTax = _buyTax + _sellTax;  
    require(buyTax <= 10, "buy Fees cannot exceed 10%");  
    require(sellTax <= 10, "buy Fees cannot exceed 10%");  
    emit UpdateTax(buyTax, sellTax);  
}
```



AUDIT SUMMARY

Project name -Twitter X

Date: 25 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	1	0	0	0
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/address/0xF28319f77103aF64F210c5593B98C033CB70A027>



Token Information

Token Name : Twitter X

Token Symbol: Twitter X

Decimals: 18

Token Supply: 10,000,000,000

Token Address:

0x17f3615586795015Ca0cf5dCd755D01b48A964a5

Checksum:

050f6e58d9f2fc3bdf49e48c8a8165aa502e7f51

Owner:

0x4bF03c20896802cC5D72FAC49998D40679A2D0fC

(at time of writing the audit)

Deployer:

0x4bF03c20896802cC5D72FAC49998D40679A2D0fC



TOKEN OVERVIEW

Fees:

Buy Fees: 0-49%

Sell Fees: 0-49%

Transfer Fees: 0%

Fees Privilege: owner

Ownership: owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: no

Blacklist: No

Other Privileges: Initial distribution of the tokens
modifying fees



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 is exercised 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



Return values of low-level calls



Gasless Send



Private modifier



Using block.timestamp



Multiple Sends



Re-entrancy



Using Suicide



Tautology or contradiction



Gas Limit and Loops



Timestamp Dependence



Address hardcoded



Revert/require functions



Exception Disorder



Use of tx.origin



Using inline assembly



Integer overflow/underflow



Divide before multiply



Dangerous strict equalities



Missing Zero Address Validation



Using SHA3



Compiler version not fixed



Using throw



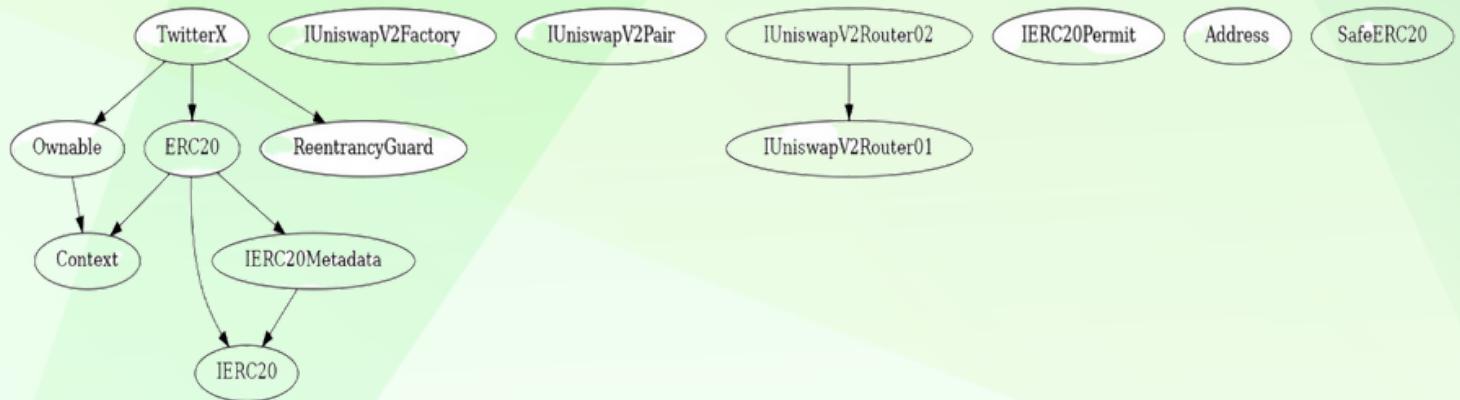
CLASSIFICATION OF RISK

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	1
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE





POINTS TO NOTE

- Owner is able to update buy and sell fees (0-49%)
- Owner is not able to set transfers
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to mint new tokens
- Owner is not able to set maximum wallet and maximum buy/sell limits

CONTRACT ASSESSMENT

Contract	Type	Bases		
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
L _msgSender	Internal			
L _msgData	Internal			
IUniswapV2Factory	Interface			
L feeTo	External !	NO !		
L feeToSetter	External !	NO !		
L getPair	External !	NO !		
L allPairs	External !	NO !		
L allPairsLength	External !	NO !		
L createPair	External !	NO !		
L setFeeTo	External !	NO !		
L setFeeToSetter	External !	NO !		
IUniswapV2Pair	Interface			
L name	External !	NO !		
L symbol	External !	NO !		
L decimals	External !	NO !		
L totalSupply	External !	NO !		
L balanceOf	External !	NO !		
L allowance	External !	NO !		
L approve	External !	NO !		
L transfer	External !	NO !		
L transferFrom	External !	NO !		
L DOMAIN_SEPARATOR	External !	NO !		
L PERMIT_TYPEHASH	External !	NO !		
L nonces	External !	NO !		
L permit	External !	NO !		
L MINIMUM_LIQUIDITY	External !	NO !		

CONTRACT ASSESSMENT

```

| L | factory | External ! | NO ! | |
| L | token0 | External ! | NO ! |
| L | token1 | External ! | NO ! |
| L | getReserves | External ! | NO ! |
| L | price0CumulativeLast | External ! | NO ! |
| L | price1CumulativeLast | External ! | NO ! |
| L | kLast | External ! | NO ! |
| L | mint | External ! | ⚡ | NO ! |
| L | burn | External ! | ⚡ | NO ! |
| L | swap | External ! | ⚡ | NO ! |
| L | skim | External ! | ⚡ | NO ! |
| L | sync | External ! | ⚡ | NO ! |
| L | initialize | External ! | ⚡ | NO ! |
|||||
| **IUniswapV2Router01** | Interface | ||
| L | factory | External ! | NO ! |
| L | WETH | External ! | NO ! |
| L | addLiquidity | External ! | ⚡ | NO ! |
| L | addLiquidityETH | External ! | 💸 | NO ! |
| L | removeLiquidity | External ! | ⚡ | NO ! |
| L | removeLiquidityETH | External ! | ⚡ | NO ! |
| L | removeLiquidityWithPermit | External ! | ⚡ | NO ! |
| L | removeLiquidityETHWithPermit | External ! | ⚡ | NO ! |
| L | swapExactTokensForTokens | External ! | ⚡ | NO ! |
| L | swapTokensForExactTokens | External ! | ⚡ | NO ! |
| L | swapExactETHForTokens | External ! | 💸 | NO ! |
| L | swapTokensForExactETH | External ! | ⚡ | NO ! |
| L | swapExactTokensForETH | External ! | ⚡ | NO ! |
| L | swapETHForExactTokens | External ! | 💸 | NO ! |
| L | quote | External ! | NO ! |
| L | getAmountOut | External ! | NO ! |

```

CONTRACT ASSESSMENT

```

| L | getAmountIn | External ! | NO ! |
| L | getAmountsOut | External ! | NO ! |
| L | getAmountsIn | External ! | NO ! |
|||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 ||
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External ! | ⚡ |
NO ! |
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! | |
| ⚡ | NO ! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! |
| ⚡ | NO ! |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | ⚡ |
NO ! |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ⚡ |
NO ! |
||||| | |
| **IERC20Permit** | Interface | ||
| L | permit | External ! | ⚡ | NO ! |
| L | nonces | External ! | NO ! |
| L | DOMAIN_SEPARATOR | External ! | NO ! |
|||||
| **Address** | Library | ||
| L | isContract | Internal 🔒 | ||
| L | sendValue | Internal 🔒 | ⚡ | ||
| L | functionCall | Internal 🔒 | ⚡ | ||
| L | functionCall | Internal 🔒 | ⚡ | ||
| L | functionCallWithValue | Internal 🔒 | ⚡ | ||
| L | functionCallWithValue | Internal 🔒 | ⚡ | ||
| L | functionStaticCall | Internal 🔒 | ||
| L | functionStaticCall | Internal 🔒 | ||
| L | functionDelegateCall | Internal 🔒 | ⚡ | ||
| L | functionDelegateCall | Internal 🔒 | ⚡ | ||

```

CONTRACT ASSESSMENT

```

| L | verifyCallResultFromTarget | Internal 🔒 | || |
| L | verifyCallResult | Internal 🔒 | ||
| L | _revert | Private 🔑 | ||
|||||
| **IERC20** | Interface | ||
| L | totalSupply | External ! | |NO ! |
| L | balanceOf | External ! | |NO ! |
| L | transfer | External ! | ⚡ |NO ! |
| L | allowance | External ! | |NO ! |
| L | approve | External ! | ⚡ |NO ! |
| L | transferFrom | External ! | ⚡ |NO ! |
|||||
| **SafeERC20** | Library | ||
| L | safeTransfer | Internal 🔒 | ⚡ |||
| L | safeTransferFrom | Internal 🔒 | ⚡ |||
| L | safeApprove | Internal 🔒 | ⚡ |||
| L | safeIncreaseAllowance | Internal 🔒 | ⚡ |||
| L | safeDecreaseAllowance | Internal 🔒 | ⚡ |||
| L | safePermit | Internal 🔒 | ⚡ |||
| L | _callOptionalReturn | Private 🔑 | ⚡ |||
|||||
| **IERC20Metadata** | Interface | IERC20 ||
| L | name | External ! | |NO ! |
| L | symbol | External ! | |NO ! |
| L | decimals | External ! | |NO ! |
|||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata ||
| L | <Constructor> | Public ! | ⚡ |NO !
| L | name | Public ! | |NO ! |
| L | symbol | Public ! | |NO ! |
| L | decimals | Public ! | |NO ! |
| L | totalSupply | Public ! | |NO ! |
| L | balanceOf | Public ! | |NO !

```

CONTRACT ASSESSMENT

```

| L | transfer | Public ! | 🔴 | NO ! | |
| L | allowance | Public ! | | NO ! |
| L | approve | Public ! | 🔴 | NO ! |
| L | transferFrom | Public ! | 🔴 | NO ! |
| L | increaseAllowance | Public ! | 🔴 | NO ! |
| L | decreaseAllowance | Public ! | 🔴 | NO ! |
| L | _transfer | Internal 🔒 | 🔴 |||
| L | _mint | Internal 🔒 | 🔴 |||
| L | _burn | Internal 🔒 | 🔴 |||
| L | _approve | Internal 🔒 | 🔴 |||
| L | _spendAllowance | Internal 🔒 | 🔴 |||
| L | _beforeTokenTransfer | Internal 🔒 | 🔴 |||
| L | _afterTokenTransfer | Internal 🔒 | 🔴 |||
|||||
| **Ownable** | Implementation | Context ||
| L | <Constructor> | Public ! | 🔴 | NO ! |
| L | owner | Public ! | NO ! |
| L | _checkOwner | Internal 🔒 |||
| L | renounceOwnership | Public ! | 🔴 | onlyOwner |
| L | transferOwnership | Public ! | 🔴 | onlyOwner |
| L | _transferOwnership | Internal 🔒 | 🔴 |||
|||||
| **ReentrancyGuard** | Implementation | ||
| L | <Constructor> | Public ! | 🔴 | NO ! |
| L | _nonReentrantBefore | Private 🔒 | 🔴 |||
| L | _nonReentrantAfter | Private 🔒 | 🔴 |||
| L | _reentrancyGuardEntered | Internal 🔒 | |||
|||||
| **TwitterX** | Implementation | ERC20, Ownable, ReentrancyGuard ||
| L | <Constructor> | Public ! | 🔴 | ERC20 |
| L | <Receive Ether> | External ! | 💸 | NO ! |
| L | <Fallback> | External ! | 💸 | NO ! |

```

CONTRACT ASSESSMENT

```

| L | getRouterAddress | Public ! | NO ! | | |
| L | claimStuckTokens | External ! | 🔴 | onlyOwner |
| L | excludeFromFees | External ! | 🔴 | onlyOwner |
| L | isExcludedFromFees | Public ! | NO ! |
| L | setAutomatedMarketMakerPair | Public ! | 🔴 | onlyOwner |
| L | isAutomatedMarketMakerPair | Public ! | NO ! |
| L | updateTax | External ! | 🔴 | onlyOwner |
| L | toggleSwapBack | External ! | 🔴 | onlyOwner |
| L | setSwapTokensAtAmount | External ! | 🔴 | onlyOwner |
| L | _transfer | Internal 🔒 | 🔴 |||
| L | ForceSwapBack | External ! | 🔴 | NO ! |
| L | _autoswapBack | Internal 🔒 | 🔴 |||
| L | outBNB | Internal 🔒 | 🔴 | nonReentrant |

```

Legend

Symbol	Meaning
----- -----	
🔴	Function can modify state
💵	Function is payable



STATIC ANALYSIS

```
Pragma version^0.8.17 (contracts/Token.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#366-374):
- (success) = recipient.call{value: amount}() (contracts/Token.sol#369)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#406-419):
- (success,returnData) = target.call{value: value}(data) (contracts/Token.sol#416)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#429-437):
- (success,returnData) = target.staticcall(data) (contracts/Token.sol#434)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#451-459):
- (success,returnData) = target.delegatecall(data) (contracts/Token.sol#456)
Low level call in TwitterX.outBNB(address,uint256) (contracts/Token.sol#1129-1135):
- (success) = address._to.call{value: amount}() (contracts/Token.sol#1132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#79) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#81) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#112) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#152) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (contracts/Token.sol#358) is not in mixedCase
Parameter TwitterX.updateTax(uint256,uint256)_buyTax (contracts/Token.sol#1013) is not in mixedCase
Parameter TwitterX.updateTax(uint256,uint256)_sellTax (contracts/Token.sol#1013) is not in mixedCase
Function TwitterX.ForceSwapBack() (contracts/Token.sol#1093-1099) is not in mixedCase
Parameter TwitterX.outBNB(address,uint256)_to (contracts/Token.sol#1129) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#157) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#158)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

TwitterX.constructor() (contracts/Token.sol#925-955) uses literals with too many digits:
- _mint(owner(),100_000_00000 * (10 ** 18)) (contracts/Token.sol#926)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

TwitterX.marketingWallet (contracts/Token.sol#902) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

TwitterX.marketingWalletShares (contracts/Token.sol#904) should be immutable
TwitterX.setSwapTokensLimit (contracts/Token.sol#907) should be immutable
TwitterX.taxDenominator (contracts/Token.sol#897) should be immutable
TwitterX.uniswapV2Pair (contracts/Token.sol#912) should be immutable
TwitterX.uniswapV2Router (contracts/Token.sol#911) 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/0xcf34bffc5ff4100ef7293f30effc2bd8c5f6c337f6412a9661f33b27e594afe9>

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

<https://testnet.bscscan.com/tx/0xcf34bffc5ff4100ef7293f30effc2bd8c5f6c337f6412a9661f33b27e594afe9>

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

<https://testnet.bscscan.com/tx/0xf09920168481f87948e6293a78306bb86f42a9214d246651fa91dbed2e32f9c0>

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

<https://testnet.bscscan.com/tx/0x0f4604da2d8db0ae6641efc0f497b7beec8bdb1ee0bb91c6acda368e3d29d503>

5- Buying when not excluded from fees (0-49% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xf6a3f93278b7c7f10f14ea11890c77d2bbf6922fb78d26d4e6d516f21d1d76f>

6- Selling when not excluded from fees (0-49% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x2bc49ba964095521f7fae912acbd003b3edfe098b4f8415877930538cc65633e>



FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xaa60c229c0da63ffef7ca3614a2c52b304e0432f1747daeb4b2513b6f17933b1>

8- Internal swap (passed):

<https://testnet.bscscan.com/address/0xb88c7e1807a2cb38c398f685e20096cac6a5a519#internaltx>



MANUAL TESTING

Centralization – Excessive fees

Severity: **High**

function: updateTax

Status: Not Resolved

Overview:

Owner is able to put upto 50% tax on buy and sells separately.

```
function updateTax(uint256 _buyTax, uint256 _sellTax) external onlyOwner {  
    buyTax = _buyTax;  
    sellTax = _sellTax;  
    totalTax = _buyTax + _sellTax;  
    require(buyTax <= 49, "buy Fees cannot exceed 49%");  
    require(sellTax <= 49, "buy Fees cannot exceed 49%");  
    emit UpdateTax(buyTax, sellTax);  
}
```

Suggestion

Consider reducing the upper bound of maximum buy/sell tax. Usually 10% is suggested to be a reasonable upperbound limit for buy/sell/transfer tax.

```
function updateTax(uint256 _buyTax, uint256 _sellTax) external onlyOwner {  
    buyTax = _buyTax;  
    sellTax = _sellTax;  
    totalTax = _buyTax + _sellTax;  
    require(buyTax <= 10, "buy Fees cannot exceed 10%");  
    require(sellTax <= 10, "buy Fees cannot exceed 10%");  
    emit UpdateTax(buyTax, sellTax);  
}
```



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 specialize 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>
