



Smart Contract Audit

FOR
Xi15
DATED : 6 august 23'



MANUAL TESTING

Centralization – Enabling Trades

Severity: High

function: enableTrading

Status: Open

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens (even owner of whitelisted wallets)

```
function enableTrading() external onlyOwner {  
    require(!tradingEnabled, "Cannot re-enable trading");  
    tradingEnabled = true;  
    providingLiquidity = true;  
}
```

Suggestion

It's suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guarantee enabling of trades.



AUDIT SUMMARY

Project name -Xi15

Date: 6 august, 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	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/0x87B0927818c4080493da29d9eFc007F21aE8d79B>



Token Information

Token Name : Xi15

Token Symbol: Xi15

Decimals: 18

Token Supply: 100,000,000

Token Address:

0x6b37970E7B1f6C27dFEd5C0ea87F480B61CEa21e

Checksum:

9e42ef4376f6db3ad6de9f8b5cd2691cbc43573c

Owner:

0x668aCA5E9fFbAD1652676341C834906577D2fC16

(at time of writing the audit)

Deployer:

0x155B6f8874b14Fece2Ed79Ba24C578A67A3B0a38



TOKEN OVERVIEW

Fees:

Buy Fees: 0-10%

Sell Fees: 0-10%

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
enabling trades
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



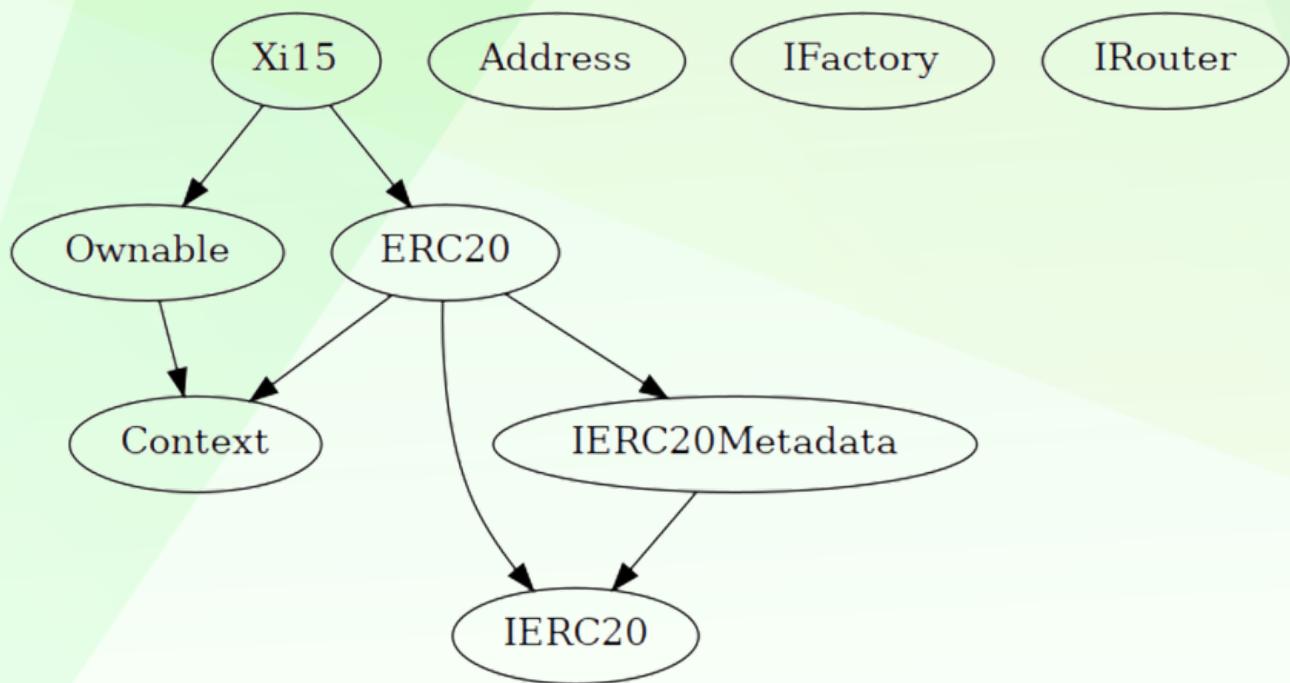
CLASSIFICATION OF RISK

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	1

INHERITANCE TREE





POINTS TO NOTE

- Owner is able to update buy/sell fees within 0-10%
- Owner is not able to set fee on transfers
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to mint new tokens
- Owner is not able to set max buy/sell/transfer
- **Owner must enable trading for investors**

CONTRACT ASSESSMENT

Contract	Type	Bases			
L **Function Name**	**Visibility**	**Mutability**	**Modifiers**		
Context	Implementation				
L _msgSender	Internal	🔒			
L _msgData	Internal	🔒			
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 !	
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 !		

CONTRACT ASSESSMENT

```

| L| decimals | Public ! | NO ! | | |
| L| totalSupply | Public ! | NO ! |
| L| balanceOf | Public ! | NO ! |
| 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| _tokengeneration | Internal 🔒 | 🔴 |||
| L| _approve | Internal 🔒 | 🔴 |||
|||||
| **Address** | Library | ||
| L| sendValue | Internal 🔒 | 🔴 |||
|||||
| **Ownable** | Implementation | Context ||
| L| <Constructor> | Public ! | 🔴 | NO ! |
| L| owner | Public ! | NO ! |
| L| renounceOwnership | Public ! | 🔴 | onlyOwner |
| L| transferOwnership | Public ! | 🔴 | onlyOwner |
| L| _setOwner | Private 🔒 | 🔴 |||
|||||
| **IFactory** | Interface | ||
| L| createPair | External ! | 🔴 | NO ! |
|||||
| **IRouter** | Interface | ||
| L| factory | External ! | NO ! |
| L| WETH | External ! | NO ! |
| L| addLiquidityETH | External ! | 💸 | NO ! |
| L| swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | 🔴 | NO ! |
|||||
| **Xi15** | Implementation | ERC20, Ownable ||
| L| <Constructor> | Public ! | 🔴 | ERC20 |
| L| approve | Public ! | 🔴 | NO ! |
| L| transferFrom | Public ! | 🔴 | NO ! |
| L| increaseAllowance | Public ! | 🔴 | NO ! |
| L| decreaseAllowance | Public ! | 🔴 | NO !

```

CONTRACT ASSESSMENT

	L transfer Public	!			NO !	
	L _transfer Internal					
	L Liquify Private				lockTheSwap	
	L swapTokensForETH Private					
	L addLiquidity Private					
	L setLiquidityProvide External	!			onlyOwner	
	L setLiquidityTreshhold External	!			onlyOwner	
	L SetBuyTaxes External	!			onlyOwner	
	L SetSellTaxes External	!			onlyOwner	
	L setMarketingWallet External	!			onlyOwner	
	L setDevelopmentWallet External	!			onlyOwner	
	L enableTrading External	!			onlyOwner	
	L excludeFromFee External	!			onlyOwner	
	L includeFromFee External	!			onlyOwner	
	L rescueBNB External	!			NO !	
	L rescueBEP20 External	!			onlyOwner	
	L <Receive Ether> External	!			NO !	

Legend

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



STATIC ANALYSIS

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Xi15.excludeFromFee(address) (contracts/Token.sol#724-728) compares to a boolean constant:
    -require(bool,string)(exemptFee[_account] != true,Account is already excluded) (contracts/Token.sol#725)
Xi15.includeFromFee(address) (contracts/Token.sol#730-734) compares to a boolean constant:
    -require(bool,string)(exemptFee[_account] != false,Account is already included) (contracts/Token.sol#731)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Context._msgData() (contracts/Token.sol#14-17) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.17 (contracts/Token.sol#7) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#338-349):
    - (success) = recipient.call{value: amount}() (contracts/Token.sol#344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Variable ERC20._balances (contracts/Token.sol#70) is not in mixedCase
Variable ERC20._allowances (contracts/Token.sol#72) is not in mixedCase
Function IRouter.WETH() (contracts/Token.sol#402) is not in mixedCase
Event Xi15.includeFromFeeUpdated(address) (contracts/Token.sol#456) is not in CapWords
Function Xi15.Liquify(uint256,Xi15.Taxes) (contracts/Token.sol#596-636) is not in mixedCase
Parameter Xi15.setLiquidityTreshold(uint256).new_amount (contracts/Token.sol#668) is not in mixedCase
Function Xi15.SetBuyTaxes(uint256,uint256,uint256) (contracts/Token.sol#680-690) is not in mixedCase
Parameter Xi15.SetBuyTaxes(uint256,uint256,uint256)._marketing (contracts/Token.sol#681) is not in mixedCase
Parameter Xi15.SetBuyTaxes(uint256,uint256,uint256)._liquidity (contracts/Token.sol#682) is not in mixedCase
Parameter Xi15.SetBuyTaxes(uint256,uint256,uint256)._development (contracts/Token.sol#683) is not in mixedCase
Function Xi15.SetSellTaxes(uint256,uint256,uint256) (contracts/Token.sol#692-702) is not in mixedCase
Parameter Xi15.SetSellTaxes(uint256,uint256,uint256)._marketing (contracts/Token.sol#693) is not in mixedCase
Parameter Xi15.SetSellTaxes(uint256,uint256,uint256)._liquidity (contracts/Token.sol#694) is not in mixedCase
Parameter Xi15.SetSellTaxes(uint256,uint256,uint256)._development (contracts/Token.sol#695) is not in mixedCase
Parameter Xi15.setMarketingWallet(address)._newWallet (contracts/Token.sol#704) is not in mixedCase
Parameter Xi15.setDevelopmentWallet(address)._newWallet (contracts/Token.sol#711) is not in mixedCase
Parameter Xi15.excludeFromFee(address)._account (contracts/Token.sol#724) is not in mixedCase
Parameter Xi15.includeFromFee(address)._account (contracts/Token.sol#730) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (contracts/Token.sol#15)" inContext (contracts/Token.sol#9-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Xi15.setLiquidityTreshold(uint256) (contracts/Token.sol#668-678) uses literals with too many digits:
    - require(bool,string)(new_amount <= 1000000,Swap threshold amount should be lower or equal to 1% of tokens) (contracts/Token.sol#669-672)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
Xi15.pair (contracts/Token.sol#428) should be immutable
Xi15.router (contracts/Token.sol#427) 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/0xb73d7d8f91e7c9745a404dec0dcf8d7fccb7e23e34cca7bcf5787643bdc576a9>

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

<https://testnet.bscscan.com/tx/0xa9862d6aacao1dcd2a930502e9995c56898ca8ca37129ee763b77822bab20>

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

<https://testnet.bscscan.com/tx/0x975beb46545e847c4534cd9b54b7199e4d3ea2f6d7b4451eb3c83ef3592fbff7>

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

<https://testnet.bscscan.com/tx/0x9ee985f4a1c55c1e033f27b69cdf10ccf2dc68b0a2b5f77ccee14baab39fd6b7>

5- Buying when not excluded from fees (0-10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xd142577a3f597adfedfcc04c15867905ba1b811ca08bdb990d25c679a2fc98b7>

6- Selling when not excluded from fees (0-10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x62e660a1c759e88e0e479b17f14401a646bdb71f59a338f3246acac4f7dad2fb>



FUNCTIONAL TESTING

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

<https://testnet.bscscan.com/tx/0xb630c31dc7091e48e75016e3f7c096198a29f1dc7d137d918ad7fdb0db7cf5de>

8- Internal swap (BNB sent to marketing wallet) (passed):

<https://testnet.bscscan.com/tx/0x62e660a1c759e88e0e479b17f14401a646bdb71f59a338f3246acac4f7dad2fb>



MANUAL TESTING

Centralization – Enabling Trades

Severity: High

function: enableTrading

Status: Open

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens (even owner of whitelisted wallets)

```
function enableTrading() external onlyOwner {  
    require(!tradingEnabled, "Cannot re-enable trading");  
    tradingEnabled = true;  
    providingLiquidity = true;  
}
```

Suggestion

It's suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guarantee enabling of trades.



MANUAL TESTING

Logical – Incorrect fee configuration

Severity: **Informational**

function: SetBuyFees + SetSellFees

Status: **Open**

Overview:

Fees are not being updated correctly in SetBuyFees and SetSellFees functions. Development fees are being updated by new liquidity fee.

```
struct Taxes {  
    uint256 marketing;  
    uint256 development;  
    uint256 liquidity;  
}  
  
function SetBuyTaxes(  
    uint256 _marketing,  
    uint256 _liquidity,  
    uint256 _development  
) external onlyOwner {  
    taxes = Taxes(_marketing, _liquidity, _development);  
    require(  
        (_marketing + _liquidity + _development) <= 10,  
        "Must keep fees at 10% or less"  
    );  
}
```

Suggestion

Correct order of fees

```
function SetBuyTaxes(  
    uint256 _marketing,  
    uint256 _liquidity,  
    uint256 _development  
) external onlyOwner {  
    taxes = Taxes(_marketing, _development, _liquidity);  
    require(  
        (_marketing + _liquidity + _development) <= 10,  
        "Must keep fees at 10% or less"  
    );  
}
```



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>
