AUDIT REPORT
DATE APRIL 11TH
FOR

Oxbull
Charging the Future of Blockchains
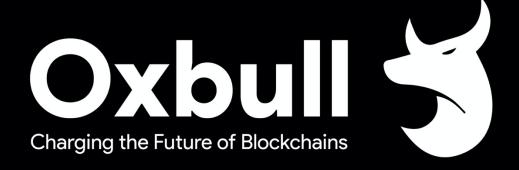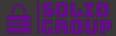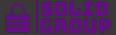
Disclaimer

SolidGroup reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team.

SolidGroup Audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors. SolidGroup Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort. Solid group audits do not cover integration with external services (such as Unicrypt, Uniswap, PancakeSawp and etc'..). SolidGroup Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidGroup's position is that each company and individual are responsible for their own due diligence and continuous security. SolidGroup in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Overview

Network: Binance Smart Chain
Website: https://www.oxbull.tech/#/home
Twitter Group: https://twitter.com/Oxbull5
Telegram Group: https://t.me/Oxbull_tech

Description

A Startup Incubator and Launchpad developed By Tech Enthusiasts. Oxbull.Tech hopes to help Startups with the necessary funding and support to kick start their project in the most efficient way possible

Files In Scope

Contract Name	Contract Address	MD5
OXBULL.sol	0x3fd5B5746315E3F8d43A46b09c826a001EBb977d (BSC)	c908603626a9e3ea74be83c4b4dda9aa

Vulnerability Summary

● Informational severity Issues	2
● Low severity issues	0
● Medium severity issues	0
● High severity issues	0

Total Issues

Informational 100%

Informational
Low
Medium
High

UML

OXBULL.sol

```
                                    ┌──────────────────────────────┐
                                    │         <<Abstract>>         │
                                    │           Context            │
                                    ├──────────────────────────────┤
                                    ├──────────────────────────────┤
                                    │ Internal:                    │
                                    │ _msgSender(): (payable: address) │
                                    │ _msgData(): bytes            │
                                    └──────────────────────────────┘
```

<<Interface>> IERC20

External:
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(recipient: address, amount: uint256): bool
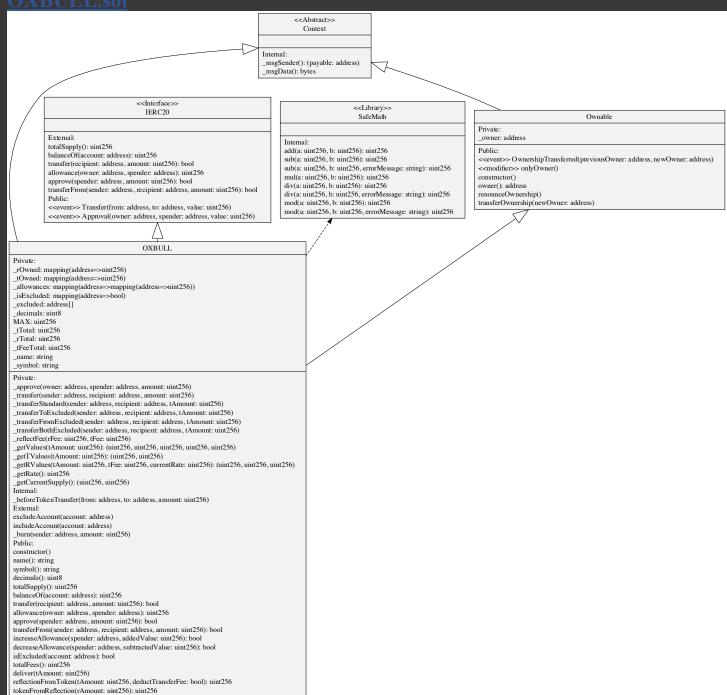allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(sender: address, recipient: address, amount: uint256): bool
Public:
<<event>> Transfer(from: address, to: address, value: uint256)
<<event>> Approval(owner: address, spender: address, value: uint256)

<<Library>> SafeMath

Internal:
add(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256, errorMessage: string): uint256
mul(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256, errorMessage: string): uint256
mod(a: uint256, b: uint256): uint256
mod(a: uint256, b: uint256, errorMessage: string): uint256

Ownable

Private:
_owner: address

Public:
<<event>> OwnershipTransferred(previousOwner: address, newOwner: address)
<<modifier>> onlyOwner()
constructor()
owner(): address
renounceOwnership()
transferOwnership(newOwner: address)

OXBULL

Private:
_rOwned: mapping(address=>uint256)
_tOwned: mapping(address=>uint256)
_allowances: mapping(address=>mapping(address=>uint256))
_isExcluded: mapping(address=>bool)
_excluded: address[]
_decimals: uint8
MAX: uint256
_tTotal: uint256
_rTotal: uint256
_tFeeTotal: uint256
_name: string
_symbol: string

Private:
_approve(owner: address, spender: address, amount: uint256)
_transfer(sender: address, recipient: address, amount: uint256)
_transferStandard(sender: address, recipient: address, tAmount: uint256)
_transferToExcluded(sender: address, recipient: address, tAmount: uint256)
_transferFromExcluded(sender: address, recipient: address, tAmount: uint256)
_transferBothExcluded(sender: address, recipient: address, tAmount: uint256)
_reflectFee(rFee: uint256, tFee: uint256)
_getValues(tAmount: uint256): (uint256, uint256, uint256, uint256, uint256)
_getTValues(tAmount: uint256): (uint256, uint256)
_getRValues(tAmount: uint256, tFee: uint256, currentRate: uint256): (uint256, uint256, uint256)
_getRate(): uint256
_getCurrentSupply(): (uint256, uint256)
Internal:
_beforeTokenTransfer(from: address, to: address, amount: uint256)
External:
excludeAccount(account: address)
includeAccount(account: address)
_burn(sender: address, amount: uint256)
Public:
constructor()
name(): string
symbol(): string
decimals(): uint8
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(recipient: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(sender: address, recipient: address, amount: uint256): bool
increaseAllowance(spender: address, addedValue: uint256): bool
decreaseAllowance(spender: address, subtractedValue: uint256): bool
isExcluded(account: address): bool
totalFees(): uint256
deliver(tAmount: uint256)
reflectionFromToken(tAmount: uint256, deductTransferFee: bool): uint256
tokenFromReflection(rAmount: uint256): uint256
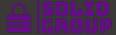
BEP-20's Conformance

This test checks for BEP-20's conformance.
- All the functions are present
- All the events are present
- Functions return the correct type
- Functions that must be view are view
- Events' parameters are correctly indexed
- The functions emit the events
- Derived contracts do not break the conformance

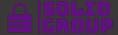| Function	present	type	Correct Return value	events	
totalSupply	✅	✅ view	✅		
balanceOf(address)	✅	✅ view	✅		
transfer(address,uint256)	✅	✅ external	✅	✅ Transfer	
transferFrom(address, address, uint256)	✅	✅ external	✅	✅ Transfer	
approve(address,uint256)	✅	✅ external	✅	✅ Approval	
allowance(address, address)	✅	✅ view	✅		
name	✅	✅ view	✅		
symbol	✅	✅ view	✅		

Check Events:
✅ Transfer
✅ Approve

General:
✅ No external mint function
✅ No Volatile Code

The contract that was tested is the token's contract: Oxbull.sol

Findings

0Xbull.sol

Description: Basic RFI contract.

Issue #1:

Type	Severity	Location
Gas Optimization	● Informational	Oxbull.sol

Description:
    The public isExcluded, totalFees, deliver, and reflectionFromToken functions should be declared as external.

Recommendation:
    These functions are only called outside the contract consider using external attribute instead of public.

Issue #2:

Type	Severity	Location
Gas Optimization	● Informational	Oxbull.sol

Description:
    State _decimals, _name, _symbol variables that are never changed should be declared as constant.

Recommendation:
    Consider using "constant" attribute for these variables.

Summary

● Informational severity Issues	2
● Low severity issues	0
● Medium severity issues	0
● High severity issues	0

s