Cyberscope

Audit Report

Quidax Token

February 2025

Network	ETH
Address	0x1780933e83b09371cf716f3630fe5a422a66a39e
Audited by	© cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	IDI	Immutable Declaration Improvement	Unresolved
●	RCT	Redundant Code Template	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L17	Usage of Solidity Assembly	Unresolved

Table of Contents

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
● Critical	Highly Likely / High Impact
● Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
● Minor / Informative	Unlikely / Low to no Impact

Review

Contract Name	QuidaxToken
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Explorer	https://etherscan.io/address/0x1780933e83b09371cf716f3630fe5a422a66a39e
Address	0x1780933e83b09371cf716f3630fe5a422a66a39e
Network	ETH
Symbol	QDX
Decimals	18
Total Supply	500.000.000

Audit Updates

Initial Audit	18 Feb 2025

Source Files

Filename	SHA256
contracts/ReflectiveERC20.sol	8e9820a2678789110e1fbad978c1c86ae642cc129f2b9bcea6766792ccdd02fd
contracts/QuidaxToken.sol	09685a538e69dbfca14670d53e689a52fa3e2368c29b10e8083de3539eed512d
contracts/lib/LibCommon.sol	ad40e79524942f0927be19739e7c96b7a52147f5cf54fdd7eb676720db70b66a

Findings Breakdown

4
Critical 0
Medium 0
Minor / Informative 4

Severity	Unresolved	Acknowledged	Resolved	Other
● Critical	0	0	0	0
● Medium	0	0	0	0
● Minor / Informative	4	0	0	0

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	contracts/QuidaxToken.sol#L150
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
maxTotalSupply
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

RCT - Redundant Code Template

Criticality	Minor / Informative
Location	contracts/ReflectiveERC20.sol contracts/QuidaxToken.sol contracts/lib/LibCommon.sol
Status	Unresolved

Description

The contract is built using an `ERC20` template designed to support multiple configurations. Specifically, during deployment, the `configProps` variable holds the configuration properties for the token contract. However, the deployed contract is configured to utilize only the `_isDocumentAllowed` property, leaving many variables, events, and functions in the contract redundant and unused. This redundancy adds unnecessary complexity to the contract, increasing both the gas costs associated with deployment and the ongoing transaction costs for users.

```
// Example of redundant variable
uint256 private constant MAX_ALLOWED_BPS = 2_000;
// Example of redundant event
event MaxTokenAmountPerSet(uint256 newMaxTokenAmount);
// Example of redundant function
function setMaxTokenAmountPerAddress(
uint256 newMaxTokenAmount
) external onlyOwner {
    if (!isMaxAmountOfTokensSet()) {
        revert MaxTokenAmountNotAllowed();
    }
    if (newMaxTokenAmount <= maxTokenAmountPerAddress) {
        revert MaxTokenAmountPerAddrLtPrevious();
    }
    maxTokenAmountPerAddress = newMaxTokenAmount;
    emit MaxTokenAmountPerSet(newMaxTokenAmount);
}
// Example of function that costs extra gas
  function transfer(
    address to,
    uint256 amount
  ) public virtual override returns (bool) {
    uint256 taxAmount = _taxAmount(msg.sender, amount);
    uint256 deflationAmount = _deflationAmount(amount);
    uint256 amountToTransfer = amount - taxAmount -
deflationAmount;
    //...
    return super.transfer(to, amountToTransfer);
  }
```

Recommendation

It is recommended to remove redundant code segments or utilize streamlined templates, such as OpenZeppelin’s `ERC20` more efficiently, to ensure the contract remains efficient and easy to maintain.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/ReflectiveERC20.sol#L38,221 contracts/QuidaxToken.sol#L160,266,280,281,300
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
function _tTotal() public view virtual returns (uint256) {
    return totalSupply();
  }
uint256 _amount
address _taxAddress
uint256 _feeBPS
uint256 _taxBPS
uint256 _deflationBPS
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/stable/style-guide.html#naming-conventions.

L17 - Usage of Solidity Assembly

Criticality	Minor / Informative
Location	contracts/lib/LibCommon.sol#L27,43,79,108
Status	Unresolved

Description

Using assembly can be useful for optimizing code, but it can also be error-prone. It's important to carefully test and debug assembly code to ensure that it is correct and does not contain any errors.

Some common types of errors that can occur when using assembly in Solidity include Syntax, Type, Out-of-bounds, Stack, and Revert.

```
assembly {
  // Transfer the ETH and check if it succeeded or not.
  if iszero(call(gas(), to, amount, 0, 0, 0, 0)) {
    // Store the function selector of `ETHTransferFailed()`.
    // bytes4(keccak256(bytes("ETHTransferFailed()"))) = 0xb12d13eb
    mstore(0x00, 0xb12d13eb)
    // Revert with (offset, size).
    revert(0x1c, 0x04)
  }
}
//...
assembly {
  let returndata_size := mload(data)
  revert(add(32, data), returndata_size)
}
```

Recommendation

It is recommended to use assembly sparingly and only when necessary, as it can be difficult to read and understand compared to Solidity code.

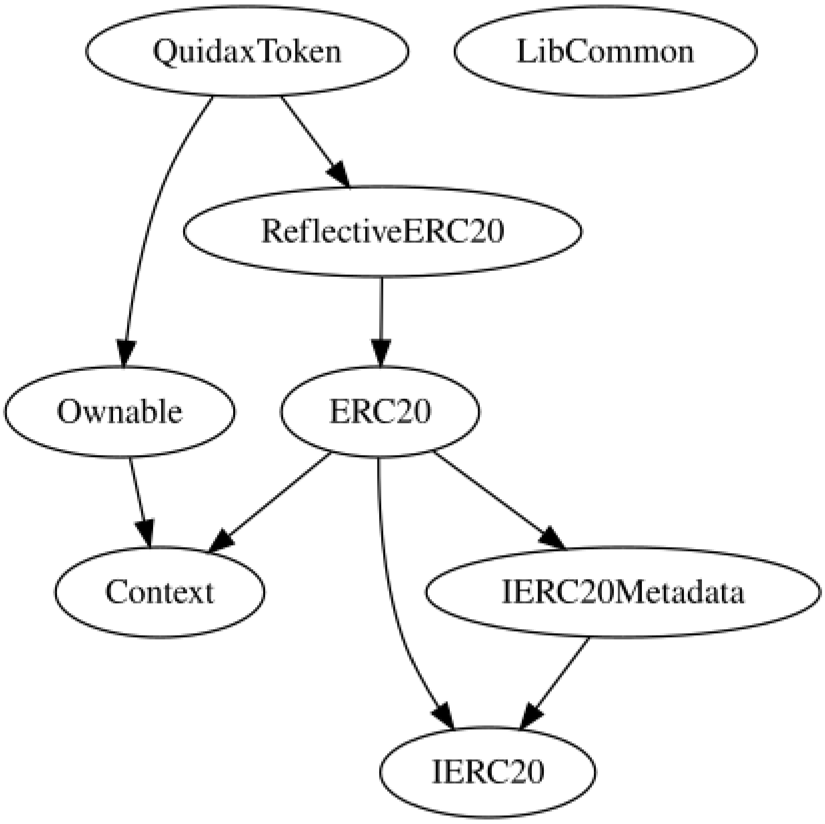
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
ReflectiveERC20	Implementation	ERC20		
	_tTotal	Public		-
		Public	✓	ERC20
	balanceOf	Public		-
	transferFrom	Public	✓	-
	transfer	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setReflectionFee	Internal	✓	
	tokenFromReflection	Public		-
	_transferReflected	Private	✓	
	_reflectFee	Private	✓	
	calculateFee	Private		
	_transferNonReflectedTax	Internal	✓	
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_rUpdate	Private	✓	

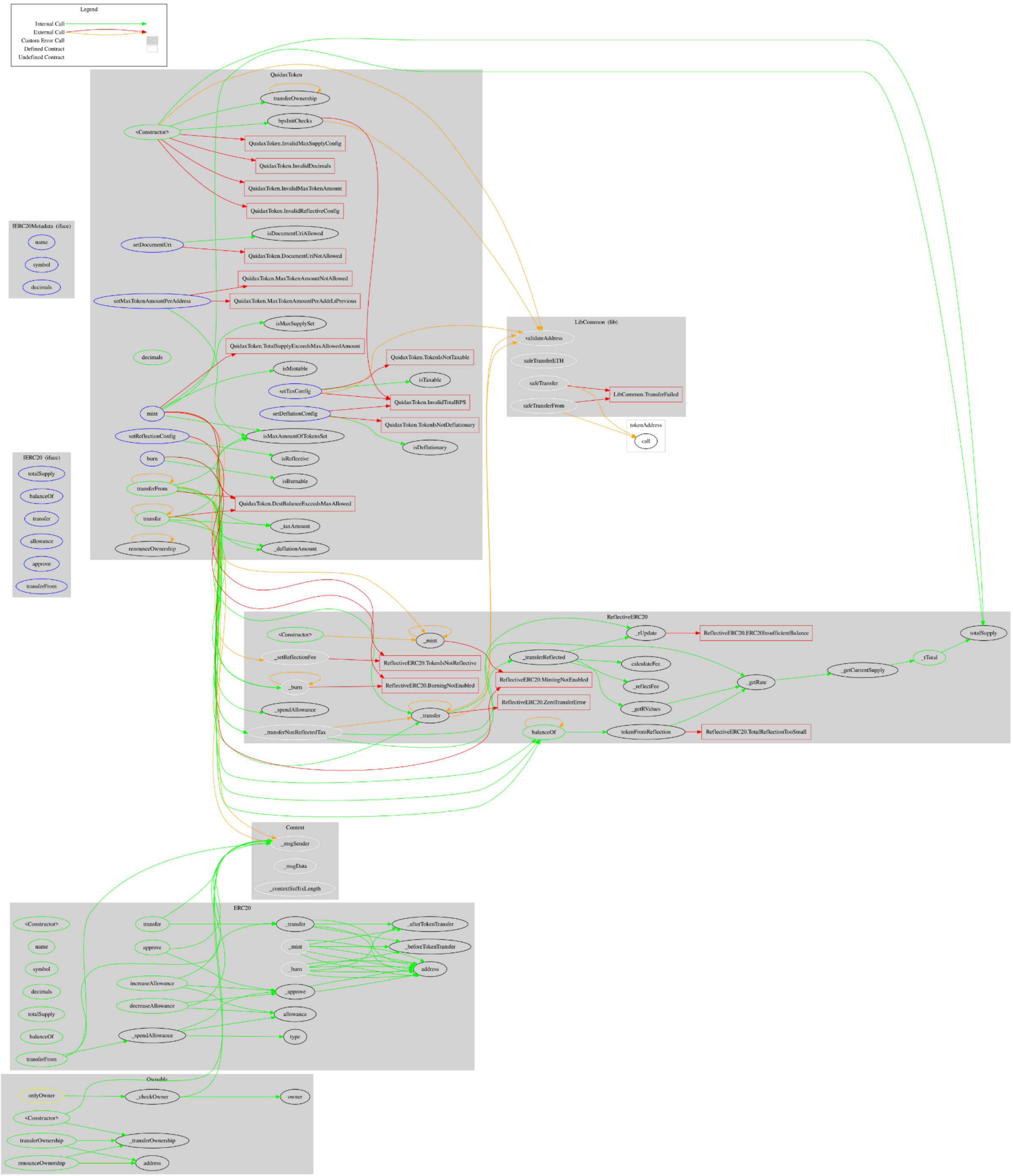
QuidaxToken	Implementation	ReflectiveERC20, Ownable		
		Public	✓	ReflectiveERC20
	bpsInitChecks	Private		
	isMintable	Public		-
	isBurnable	Public		-
	isMaxAmountOfTokensSet	Public		-
	isMaxSupplySet	Public		-
	isDocumentUriAllowed	Public		-
	decimals	Public		-
	isTaxable	Public		-
	isDeflationary	Public		-
	isReflective	Public		-
	setDocumentUri	External	✓	onlyOwner
	setMaxTokenAmountPerAddress	External	✓	onlyOwner
	setReflectionConfig	External	✓	onlyOwner
	setTaxConfig	External	✓	onlyOwner
	setDeflationConfig	External	✓	onlyOwner
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	mint	External	✓	onlyOwner
	burn	External	✓	onlyOwner
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_taxAmount	Internal		

	_deflationAmount	Internal		
LibCommon	Library			
	safeTransferETH	Internal	✓	
	validateAddress	Internal		
	safeTransferFrom	Internal	✓	
	safeTransfer	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
QuidaxToken
transferOwnership
bpsInitChecks
<Constructor>
QuidaxToken.InvalidMaxSupplyConfig
QuidaxToken.InvalidDecimals
QuidaxToken.InvalidMaxTokenAmount
QuidaxToken.InvalidReflectiveConfig
isDocumentUriAllowed
setDocumentUri
QuidaxToken.DocumentUriNotAllowed
QuidaxToken.MaxTokenAmountNotAllowed
setMaxTokenAmountPerAddress
QuidaxToken.MaxTokenAmountPerAddrLtPrevious
isMaxSupplySet
QuidaxToken.TotalSupplyExceedsMaxAllowedAmount
decimals
QuidaxToken.TokenIsNotTaxable
isMintable
isTaxable
setTaxConfig
QuidaxToken.InvalidTotalBPS
mint
setDeflationConfig
QuidaxToken.TokenIsNotDeflationary
setReflectionConfig
isMaxAmountOfTokensSet
isDeflationary
burn
isReflective
transferFrom
isBurnable
QuidaxToken.DestBalanceExceedsMaxAllowed
transfer
_taxAmount
renounceOwnership
_deflationAmount
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
LibCommon (lib)
validateAddress
safeTransferETH
safeTransfer
LibCommon.TransferFailed
safeTransferFrom
tokenAddress
call
ReflectiveERC20
<Constructor>
_mint
_rUpdate
ReflectiveERC20.ERC20InsufficientBalance
totalSupply
_setReflectionFee
ReflectiveERC20.TokenIsNotReflective
_transferReflected
calculateFee
_tTotal
_burn
ReflectiveERC20.BurningNotEnabled
ReflectiveERC20.MintingNotEnabled
_reflectFee
_getRate
_getCurrentSupply
_spendAllowance
_transfer
ReflectiveERC20.ZeroTransferError
_getRValues
_transferNonReflectedTax
balanceOf
tokenFromReflection
ReflectiveERC20.TotalReflectionTooSmall
Context
_msgSender
_msgData
_contextSuffixLength
ERC20
<Constructor>
transfer
_transfer
_afterTokenTransfer
name
approve
_mint
_beforeTokenTransfer
symbol
_burn
address
decimals
increaseAllowance
_approve
totalSupply
decreaseAllowance
allowance
balanceOf
_spendAllowance
type
transferFrom
Ownable
onlyOwner
_checkOwner
owner
<Constructor>
transferOwnership
_transferOwnership
renounceOwnership
address

Summary

Quidax Token contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Quidax Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. The contract does not implement any fees.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does 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' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment 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 Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

cyberscope.io