SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC Development | Marketing

MADE IN GERMANY

Unicrypt

\-

V2 ENMT TaxToken

Audit

Security Assessment
09. December, 2022

For

Disclaimer

SolidProof.io 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. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc'...)

SolidProof.io 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 technology proprietors. SolidProof 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.

SolidProof.io 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. SolidProof's position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	15. October 2022	· Layout project · Automated-Security Testing
	16.-17. October 2022	· Code review · Manual-Security Testing
	18.-19. October 2022	· Finishing audit
1.1	14. - 15. November 2022	· Reaudit
1.2	09. December 2022	· Small updates review · Updating report

Network
Ethereum (ERC20)

Website
https://unicrypt.network/

Telegram
https://t.me/uncx_token

Twitter
https://twitter.com/UNCX_token

Medium
https://unicrypt.medium.com/

Description

TBA

Project Engagement

During the 12th of October 2022, **Unicrypt Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

Logo

Contract Link

v1.1
- TBA

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon aspossible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:
1. Code review that includes the following:
    i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
    ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.

2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Used Code from other Frameworks/Smart Contracts (direct imports)

Imported packages:

Dependency / Import Path	Count
@uniswap/lib/contracts/libraries/TransferHelper.sol	4
@uniswap/v2-core/contracts/interfaces/IUniswapV2Callee.sol	1
@uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol	6
@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol	6
hardhat/console.sol	6

Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

File Name	SHA-1 Hash
contracts/interfaces/ILPWallet.sol	4e007b329c27488ced3f71696a637d13e7126e8a
contracts/interfaces/IFeeHelper.sol	60155a4698d6e2cc9b00bee3d5faf696b7d5405e
contracts/interfaces/ISettings.sol	d4d98359089dbb27a407382cc28fa0fa828bdcf6
contracts/interfaces/IFacetHelper.sol	1183a8fb60a5b336b2b9b1a4689064e992c50853
contracts/interfaces/IUniswapV2Pair.sol	ccd8036e1ef9d2430f46a45fa801fc625dd80d8e
contracts/interfaces/ILosslessController.sol	abd92ff28638a979cef802a2f43f594bf07cf906
contracts/interfaces/IUniswapV2Factory.sol	a5d78edcba4e2228f92a4a0df03190c12d869184
contracts/interfaces/IBuyBackWallet.sol	1be52eee7179671108769fbc481b047d20b28060
contracts/interfaces/ITaxToken.sol	020247b8d261aa00866a4b3d07d1ffad02ae46c6
contracts/interfaces/IMintFactory.sol	308080b7c16fd2a083c9b26e61587347c9def74c
contracts/interfaces/IUniswapV2Router02.sol	9b9f4c23ac1e66692519984e3d449605afa8a3bc
contracts/interfaces/IUniswapV2Router01.sol	fc9a0f0007cb1ba6c3f8f3e63f0fa6280d4459d4
contracts/interfaces/IWallets.sol	702fe0fd8416132422cfdb811576e2bd8d6dfb99
contracts/interfaces/ITaxHelper.sol	e207a9620c792ee973a2160161a95e4d9be43f85
contracts/interfaces/IERC20.sol	d291a1b2b60f170b82ac516d27850744443b1061
contracts/BuyBackWallet.sol	c7cc1c04f2f849fd6512f56460b46e63dc46819f
contracts/FacetHelper.sol	ac9fd8c642febdbae6a1ee86fab342abe215fbda
contracts/TaxToken.sol	289d8327bbc5f205890c8644510f564e189668d2
contracts/FeeHelper.sol	7f1f246911c9b64c76fedae5c6e82e04463b910b
contracts/libraries/FullMath.sol	f3bb311a92c67379bd610c8080c0b658401660c5
contracts/libraries/Context.sol	f5d8feec0045f865471b23db56f61350606cbf25
contracts/libraries/EnumerableSet.sol	44c136b4346bf33f65749875647195534b896d7d
contracts/libraries/Ownable.sol	2e4152d722d76a8e5ebfb039dd45c46dcea56b23
contracts/libraries/Pausable.sol	b92436672e431cb4a45d52af5a74278640402b0f
contracts/libraries/ERC20.sol	c73d52394785fbc6f42e6308b41704a06bcda8ae
contracts/MintGenerator.sol	0162d6f613fe47e71af85356553f6dca3744e0a5
contracts/MintFactory.sol	1597381e49e05e27d74f515adbaeb412fa63dd55
contracts/facets/AntiBot.sol	599fa2673630995f321dda383096de87810f8f2c
contracts/facets/Tax.sol	860214a8083f58bfec968ee0763a39f39ba421b0
contracts/facets/Constructor.sol	7e524824700477113d18d6a357cef686c05d4f9f
contracts/facets/Storage.sol	6df7751fb28b4906c00563a91202560cf998ad17
contracts/facets/Multicall.sol	4a50511c71424b60811181d8b50d8fe859c23653
contracts/facets/Lossless.sol	46bdbb0030c8e4ae1b60ba8dbbbf55511d2b13aa
contracts/facets/Wallets.sol	e39bdc224cd664a796c52600a5c1905bcbfe7c58
contracts/facets/Settings.sol	3299ced4d1ad1a1891f2bf7c9bd6ad6e8d735efe
contracts/TaxHelperUniswapV2.sol	7fad9e43710cbf08951272d4036ba4150e0f0965
contracts/LPWallet.sol	721108476da44d1a08bde7de3c9be02fd253d138

Metrics

Source Lines
v1.0
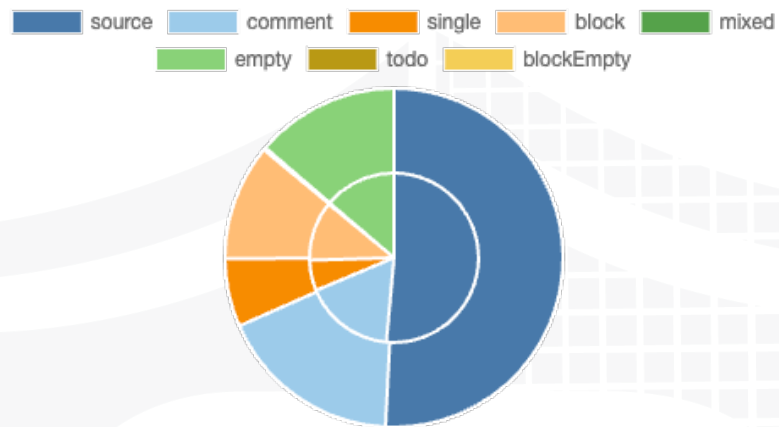
Risk Level
v1.0
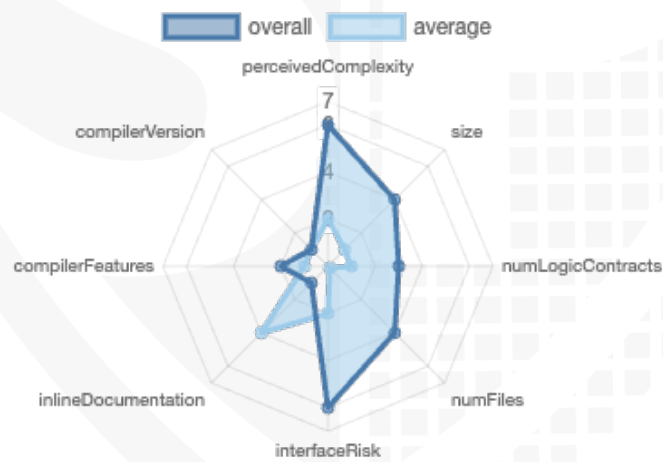
Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	17	2	15	3

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

Version	Public	Payable
1.0	300	9

Version	External	Internal	Private	Pure	View
1.0	184	252	18	13	141

State Variables

| Version	Total	Public
1.0	82	37

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	`^0.8.0` `>=0.5.0` `>=0.6.2`		yes	yes (10 asm blocks)	

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	EC Recover	New/Create/Create2
1.0	yes		yes	yes		yes → NewContract:TaxToken → NewContract:BuyBackWallet → NewContract:LPWallet

Inheritance Graph
v1.0
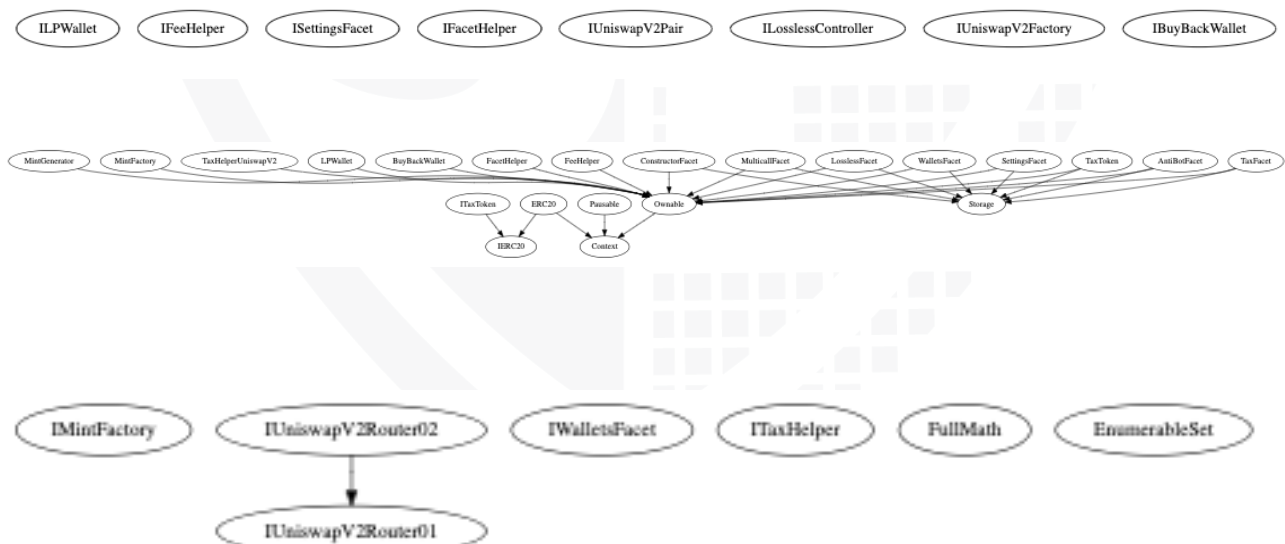
CallGraph
v1.0

Scope of Work/Verify Claims

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:
1. Is contract an upgradeable
2. Correct implementation of Token standard
3. Deployer cannot mint any new tokens
4. Deployer cannot burn or lock user funds
5. Deployer cannot pause the contract
6. Deployer cannot set fees
7. Deployer cannot blacklist/antisnipe addresses
8. Overall checkup (Smart Contract Security)

Is contract an upgradeable

Name	
Is contract an upgradeable?	**Yes**

Comments:
v1.0

- Since this contract is based on the diamond-3 pattern of Nick Mudge (For more information visit: https://eips.ethereum.org/EIPS/eip-2535, https://github.com/mudgen/diamond-3) parts of the project can be replaced by the owner. The team can also implement new functionalities after the deployment.

Correct implementation of Token standard

ERC20				
Function	**Description**	**Exist**	**Tested**	**Verified**
TotalSupply	Provides information about the total token supply	✓	✓	✓
BalanceOf	Provides account balance of the owner's account	✓	✓	✓
Transfer	Executes transfers of a specified number of tokens to a specified address	✓	✓	✓
TransferFrom	Executes transfers of a specified number of tokens from a specified address	✓	✓	✓
Approve	Allow a spender to withdraw a set number of tokens from a specified account	✓	✓	✓
Allowance	Returns a set number of tokens from a spender to the owner	✓	✓	✓

ERC721				
Function	**Description**	**Exist**	**Tested**	**Verified**
BalanceOf	Count all NFTs assigned to an owner	✓	✓	✓
OwnerOf	Find the owner of an NFT	✓	✓	✓
SafeTransferFrom	Transfers the ownership of an NFT from one address to another address	✓	✓	✓
SafeTransferFrom	See above - Difference is that this function has an extra data parameter	✓	✓	✓
TransferFrom	Transfer ownership of an NFT	✓	✓	✓
Approve	Change or reaffirm the approved address for an NFT	✓	✓	✓
SetApprovalForAll	Enable or disable approval for a third party ("operator") to manage all of `msg.sender`'s assets	✓	✓	✓
GetApproved	Get the approved address for a single NFT	✓	✓	✓
IsApprovedForAll	Query if an address is an authorized operator for another address	✓	✓	✓
SupportsInterface	Query if a contract implements an interface	✓	✓	✓
Name	Provides information about the name	✓	✓	✓
Symbol	Provides information about the symbol	✓	✓	✓
TokenURI	Provides information about the TokenUri	✓	✓	✓

Write functions of contract v1.0

AntiBot

```
setIncrement
setEndDate
setInitialMaxHold
updateAntiBot
antiBotCheck
addMaxBalanceWhitelistedAddress
removeMaxBalanceWhitelistedAddress
updateMaxBalanceWhitelistBatch
updateMaxBalanceAfterBuy
addSwapWhitelistedAddress
removeSwapWhitelistedAddress
updateSwapWhitelistBatch
setSwapWhitelistEndDate
updateSwapWhitelisting
swapWhitelistingCheck
```

Constructor

```
constructorHandler
```

Lossless

```
setLosslessAdmin
transferRecoveryAdminOwnership
acceptRecoveryAdminOwnership
proposeLosslessTurnOff
executeLosslessTurnOff
executeLosslessTurnOn
```

ERC20

```
transfer
approve
transferFrom
increaseAllowance
decreaseAllowance
```

Ownable

```
renounceOwnership
transferOwnership
```

BuyBackWallet

```
sendEthToTaxHelper
updateThreshold
```

FacetHelper

```
addFacet
addSelector
removeSelector
resetFacetStorage
updateSettingsFacet
updateLosslessFacet
updateTaxFacet
updateConstructorFacet
updateWalletsFacet
updateAntiBotFacet
updateMulticallFacet
```

Multicall

```
multicallAdminUpdate
multicallAntiBotUpdate
```

Settings

```
addLPToken
removeLPToken
togglePause
addBlacklistedAddress
removeBlacklistedAddress
updateBlacklistBatch
updateCustomTaxes
updateTaxFees
updateTransactionTaxAddress
lockSettings
updateSettings
updatePairAddress
updateTaxHelperIndex
```

Tax

```
handleTaxes
_transfer
reflect
excludeAccount
includeAccount
mint
burn
```

Wallets

```
createBuyBackWallet
createLPWallet
updateBuyBackWalletThreshold
updateLPWalletThreshold
```

FeeHelper

```
setGeneratorFee
setFee
setFeeAddress
```

LPWallet

```
sendEthToTaxHelper
transferBalanceToTaxHelper
updateThreshold
```

MintFactory

```
adminAllowTokenGenerator
addTaxHelper
updateTaxHelper
registerToken
updateFacetHelper
updateFeeHelper
updateLosslessController
```

MintGenerator

```
createToken 💰
```

TaxHelperUniswapV2

```
initiateBuyBackTax 💰
initiateLPTokenTax
createLPToken
```

TaxToken

```
transferOutBlacklistedFunds
buyBackBurn
transfer
approve
transferFrom
increaseAllowance
decreaseAllowance
```

Deployer cannot mint any new tokens

Name	Exist	Tested	Status
Deployer cannot mint	✓	✓	✗

Comments:
v1.0
- Owner can mint new tokens

Deployer cannot burn or lock user funds

Name	Exist	Tested	Status
Deployer cannot lock	✓	✓	✗
Deployer cannot burn	✓	✓	✗

Comments:
v1.0
- Owner can lock user funds by
    - blacklisting addresses
    - Settings fees too high

- Tokens
    - can be burned by the owner
    - can be burned by msg.sender
    - Will be burned in the "initialBuyBackTax" function in the TaxHelperUniswapV2 contract L41

Deployer cannot pause the contract

Name	Exist	Tested	Status
Deployer cannot pause	✓	✓	✗

Comments:
v1.0

- Owner can pause contract

Deployer cannot set fees

Name	Exist	Tested	Status
Deployer cannot set fees over 25%	✓	✓	✗
Deployer cannot set fees to nearly 100% or to 100%	✓	✓	✗

Comments:
v1.0
- FeeHelper
  - Fees can be set without any limitations

Deployer can blacklist/antisnipe addresses

Name	Exist	Tested	Status
Deployer cannot blacklist/antisnipe addresses	✓	✓	✗

Comments:
v1.0
- Only contracts can be blacklisted

Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

Legend

Attribute	Symbol
Verified / Checked	✓
Partly Verified	🚩
Unverified / Not checked	✗
Not available	–

Modifiers and public functions
v1.0

Lossless

- ⌄ ◆ setLosslessAdmin
  - ◎ onlyRecoveryAdmin
- ⌄ ◆ transferRecoveryAdminOwnership
  - ◎ onlyRecoveryAdmin
- ◆ acceptRecoveryAdminOwnership
- ⌄ ◆ proposeLosslessTurnOff
  - ◎ onlyRecoveryAdmin
- ⌄ ◆ executeLosslessTurnOff
  - ◎ onlyRecoveryAdmin
- ⌄ ◆ executeLosslessTurnOn
  - ◎ onlyRecoveryAdmin

MintFactory

- ⌄ ◆ adminAllowTokenGenerator
  - ◎ onlyOwner
- ⌄ ◆ addTaxHelper
  - ◎ onlyOwner
- ⌄ ◆ updateTaxHelper
  - ◎ onlyOwner
- ◆ registerToken
- ⌄ ◆ updateFacetHelper
  - ◎ onlyOwner
- ⌄ ◆ updateFeeHelper
  - ◎ onlyOwner
- ⌄ ◆ updateLosslessController
  - ◎ onlyOwner

MintGenerator

- ◆ createToken 💰

TaxHelperUniswapV2

- ⌄ ◆ initiateBuyBackTax 💰
  - ◎ isToken
- ⌄ ◆ initiateLPTokenTax
  - ◎ isToken
- ◆ createLPToken

TaxToken

- ◆ transferOutBlacklistedFunds
- ◆ buyBackBurn
- ◆ transfer
- ◆ approve
- ◆ transferFrom
- ◆ increaseAllowance
- ◆ decreaseAllowance

AntiBot

- setIncrement
  - onlyOwner
  - antiBotIsActive
- setEndDate
  - onlyOwner
  - antiBotIsActive
- setInitialMaxHold
  - onlyOwner
  - antiBotIsActive
- updateAntiBot
  - onlyOwner
- antiBotCheck
- addMaxBalanceWhitelistedAddress
  - onlyOwner
- removeMaxBalanceWhitelistedAddress
  - onlyOwner
- updateMaxBalanceWhitelistBatch
  - onlyOwner
- updateMaxBalanceAfterBuy
  - onlyOwner
- addSwapWhitelistedAddress
  - onlyOwner
- removeSwapWhitelistedAddress
  - onlyOwner
- updateSwapWhitelistBatch
  - onlyOwner
- setSwapWhitelistEndDate
  - onlyOwner
- updateSwapWhitelisting
  - onlyOwner
- swapWhitelistingCheck

Settings

- addLPToken
  - onlyOwner
- removeLPToken
  - onlyOwner
- togglePause
  - onlyOwner
- addBlacklistedAddress
  - onlyOwner
  - canBlacklist
- removeBlacklistedAddress
  - onlyOwner
  - canBlacklist
- updateBlacklistBatch
  - onlyOwner
  - canBlacklist
- updateCustomTaxes
  - onlyOwner
- updateTaxFees
  - onlyOwner
- updateTransactionTaxAddress
  - onlyOwner
- lockSettings
  - onlyOwner
- updateSettings
  - onlyOwner
- updatePairAddress
  - onlyOwner
- updateTaxHelperIndex
  - onlyOwner

27

FacetHelper

- addFacet
  - onlyOwner
- addSelector
  - onlyOwner
- removeSelector
  - onlyOwner
- resetFacetStorage
  - onlyOwner
- updateSettingsFacet
  - onlyOwner
- updateLosslessFacet
  - onlyOwner
- updateTaxFacet
  - onlyOwner
- updateConstructorFacet
  - onlyOwner
- updateWalletsFacet
  - onlyOwner
- updateAntiBotFacet
  - onlyOwner
- updateMulticallFacet
  - onlyOwner

Tax

- handleTaxes
- _transfer
- reflect
- excludeAccount
  - onlyOwner
- includeAccount
  - onlyOwner
- mint
  - onlyOwner
- burn

Constructor

- constructorHandler

Multicall

- multicallAdminUpdate
  - onlyOwner
- multicallAntiBotUpdate
  - onlyOwner

Wallets

- createBuyBackWallet
- createLPWallet
- ∨ updateBuyBackWalletThreshold
  - onlyOwner
- ∨ updateLPWalletThreshold
  - onlyOwner

BuyBackWallet

- sendEthToTaxHelper
- ∨ updateThreshold
  - onlyOwner

FeeHelper

- ∨ setGeneratorFee
  - onlyOwner
- ∨ setFee
  - onlyOwner
- ∨ setFeeAddress
  - onlyOwner

LPWallet

- sendEthToTaxHelper
- transferBalanceToTaxHelper
- ∨ updateThreshold
  - onlyOwner

Comments

- *Deployer can set following state variables without any limitations*
  - **AntiBot**
    - maxBalanceAfterBuy
    - antiBotSettings.initialMaxHold
    - antiBotSettings.increment
  - **Multicall**
    - fees.transactionTax.buy
    - fees.transactionTax.sell

- fees.buyBackTax
- fees.holderTax
- fees.lpTax
- antiBotSettings.increment
- antiBotSettings.initialMaxHold
- **Settings**
  - taxHelperIndex
    - Max to 2^8 - 1
- **BuyBackWallet**
  - threshold
- **FacetHelper**
  - isFacet
  - facetsList
  - selectorToFacet
  - selectorsList
- **FeeHelper**
  - SETTINGS.FEE
  - SETTINGS.GENERATOR_FEE
- **LPWallet**
  - threshold
  -

- *Deployer can enable/disable following state variables*
  - **AntiBot**
    - swapWhitelistingSettings.isActive
    - antiBotSettings.isActive
    - swapWhitelistlist
    - maxBalanceWhitelistlist
  - **Lossless**
    - isLosslessTurnOffProposed
    - isLosslessOn
  - **Multicall**
    - taxSettings.transactionTax
    - taxSettings.holderTax
    - taxSettings.buyBackTax
    - taxSettings.lpTax
    - taxSettings.canMint
    - taxSettings.canPause
    - taxSettings.canBlacklist
    - taxSettings.maxBalanceAfterBuy
    - isLocked.transactionTax
    - isLocked.holderTax
    - isLocked.buyBackTax
    - isLocked.lpTax

30

- isLocked.canMint
- isLocked.canPause
- isLocked.canBlacklist
- isLocked.maxBalanceAfterBuy
- antiBotSettings.isActive
- swapWhitelistingSettings.isActive

- **Settings**
  - taxSettings.transactionTax
  - taxSettings.holderTax
  - taxSettings.buyBackTax
  - taxSettings.lpTax
  - taxSettings.canMint
  - taxSettings.canPause
  - taxSettings.canBlacklist
  - taxSettings.maxBalanceAfterBuy
  - isLocked.transactionTax
  - isLocked.holderTax
  - isLocked.buyBackTax
  - isLocked.lpTax
  - isLocked.canMint
  - isLocked.canPause
  - isLocked.canBlacklist
  - isLocked.maxBalanceAfterBuy
  - blacklist
  - isPaused
  - lpTokens

- **Tax**
  - _isExcluded
  - _excluded

- *Deployer can set following addresses/string*
  - **Lossless**
    - recoveryAdmin
    - recoveryAdminCandidate
    - recoveryAdminKeyHash
    - admin
  - **Settings**
    - pairAddress
    - transactionTaxWallet
  - **FacetHelper**
    - facets.Multicall
    - facets.AntiBot
    - facets.Wallets
    - facets.Constructor

- facets.Settings
- facets.Lossless
- facets.Tax
- **FeeHelper**
  - SETTINGS.FEE_ADDRESS
- **MintFactory**
  - LosslessController
  - FeeHelper
  - FacetHelper
  - taxHelpersData.Address
  - taxHelpers
  - tokenGenerators

- *Existing Modifiers*
  - **TaxHelperUniswapV2**
    - isToken
  - **Antibot**
    - antiBotIsActive
  - **Lossless**
    - onlyRecoveryAdmin
  - **Settings**
    - canBlacklist
  - **Ownable**
    - onlyOwner
  - **Pausable**
    - whenNotPaused
    - whenPaused

- The project is a diamond-3 structured. That means, that the owner can upgrade contracts on the fly.
- Tax
  - Owner can mint new tokens
- Wallets
  - Anybody is able to call "createBuyBackWallet" and "createLPWallet"
    - This function returns only the address of the new created wallets but it will not set a state variable in the storage
  - Owner can update buyback/lp wallets from the facet
- MintFactory
  - Can add new taxHelper
- FacetHelper
  - Owner can reset facet storage and remove all selector and facet list

Please check if an OnlyOwner or similar restrictive modifier has been forgotten.

Source Units in Scope
v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
🔍	contracts/interfaces/ILPWallet.sol	—	1	16	5	3	—	13	—
🔍	contracts/interfaces/IFeeHelper.sol	—	1	17	4	3	—	15	—
🔍	contracts/interfaces/ISettings.sol	—	1	8	5	3	—	7	—
🔍	contracts/interfaces/IFacetHelper.sol	—	1	41	5	3	—	37	—
🔍	contracts/interfaces/IUniswapV2Pair.sol	—	1	52	7	5	—	55	—
🔍	contracts/interfaces/ILosslessController.sol	—	1	92	11	2	71	25	—
🔍	contracts/interfaces/IUniswapV2Factory.sol	—	1	17	6	4	—	17	—
🔍	contracts/interfaces/IBuyBackWallet.sol	—	1	14	5	3	—	11	—
🔍	contracts/interfaces/ITaxToken.sol	—	1	14	7	4	—	9	—
🔍	contracts/interfaces/IMintFactory.sol	—	1	36	11	8	—	27	—
🔍	contracts/interfaces/IUniswapV2Router02.sol	—	1	44	6	4	—	16	⚖️
🔍	contracts/interfaces/IUniswapV2Router01.sol	—	1	95	4	3	—	48	⚖️
🔍	contracts/interfaces/IWallets.sol	—	1	11	5	3	—	9	—
🔍	contracts/interfaces/ITaxHelper.sol	—	1	18	5	3	—	9	—
🔍	contracts/interfaces/IERC20.sol	—	1	79	28	17	58	13	☀️
📝	contracts/BuyBackWallet.sol	1	—	64	64	44	3	36	⚖️
📝	contracts/FacetHelper.sol	1	—	153	153	114	5	84	—
📝	contracts/TaxToken.sol	1	—	314	314	237	29	240	🖥️👥
📝	contracts/FeeHelper.sol	1	—	55	55	38	3	22	—
📚	contracts/libraries/FullMath.sol	1	—	126	118	57	59	104	🖥️Σ
🍯	contracts/libraries/Context.sol	1	—	26	26	10	13	1	—
📚	contracts/libraries/EnumerableSet.sol	1	—	357	357	118	196	49	🖥️
🍯	contracts/libraries/Ownable.sol	1	—	70	70	27	34	24	—
🍯	contracts/libraries/Pausable.sol	1	—	91	91	29	51	16	—
📝	contracts/libraries/ERC20.sol	1	—	305	305	90	179	73	—
📝	contracts/MintGenerator.sol	1	—	47	45	22	8	36	⚖️➕◎
📝	contracts/MintFactory.sol	1	—	181	181	108	35	78	—
📝	contracts/facets/AntiBot.sol	1	—	174	174	130	14	108	—
📝	contracts/facets/Tax.sol	1	—	361	361	285	22	187	—
📝	contracts/facets/Constructor.sol	1	—	139	139	118	10	60	—
📝	contracts/facets/Storage.sol	1	—	182	182	131	7	52	—
📝	contracts/facets/Multicall.sol	1	—	178	178	133	16	67	—
📝	contracts/facets/Lossless.sol	1	—	67	67	49	4	43	🎴
📝	contracts/facets/Wallets.sol	1	—	43	43	27	3	49	◎
📝	contracts/facets/Settings.sol	1	—	200	200	166	4	118	—
📝	contracts/TaxHelperUniswapV2.sol	1	—	117	117	93	6	106	⚖️➕
📝	contracts/LPWallet.sol	1	—	73	73	50	3	44	⚖️➕
📝📚🔍🍯	**Totals**	**22**	**15**	**3877**	**3427**	**2144**	**833**	**1908**	🖥️⚖️➕👥🎴◎☀️Σ

Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalised lines of the source unit (e.g. normalises functions spanning multiple lines)
nSLOC	normalised source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments

Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, …)

Audit Results

Critical issues

No critical issues

High issues

No high issues

Medium issues

No medium issues

Low issues

Issue	File	Type	Line	Description
#1	TaxToken	Missing Zero Address Validation (missing-zero-check)	See description	Check that the address is not zero Check following variables: - _factory L54 - Params.creator_ L65 - constructorFacetAddress L56 - feeHelper L62 - feeAddress L64 Make sure to check the above variables also in the code
#2	TaxHelperUniswapV2	Missing Zero Address Validation (missing-zero-check)	69, 70	Check that the address is not zero
#3	TaxToken	Missing Zero Address Validation (missing-zero-check)	71, 119, 187, 216, 268	Check that the address is not zero
#4	Settings	Missing Zero Address Validation (missing-zero-check)	203	Check that the address is not zero

Informational issues

Issue	File	Type	Line	Description
#1	LPWallet	Error message is missing	58	Provide an error message for require statement
#2	TaxToken	Error message is missing	72	Provide an error message for require statement
#3	AntiBot	Error message is missing	102	Provide an error message for require statement
#4	Constructor	Error message is missing	87	Provide an error message for require statement
#5	Tax	Error message is missing	45	Provide an error message for require statement
#6	FullMath	Error message is missing	34, 43, 122	Provide an error message for require statement
#7	All	NatSpec documentation missing	-	If you started to comment your code, also comment all other functions, variables etc.
#8	Lossless	Set recoveryAdminCandidate to address zero after accepting	41	Don't forget to set the "recoveryAdminCandidate" to address zero after accepting the ownership
#9	AntiBot	Visibility first	16	The visibility modifier "internal" should come before other modifiers. We recommend you to put internal before the view key here. Also the same for the "maxBalanceAfterBuyCheck" function.
#10	IUniswapV2Factory	SPDX License is missing	See description	Add a SPDX License at the top of source file
#11	IUniswapV2Pair	SPDX License is missing	See description	Add a SPDX License at the top of source file
#12	IUniswapV2Router01	SPDX License is missing	See description	Add a SPDX License at the top of source file

#13	IUniswapV2Router02	SPDX License is missing	See description	Add a SPDX License at the top of source file

Audit Comments

We recommend you to use the special form of comments (NatSpec Format, Follow link for more information https://docs.soliditylang.org/en/latest/natspec-format.html) for your contracts to provide rich documentation for functions, return variables and more. This helps investors to make clear what that variables, functions etc. do.

15. November 2022:

- We recommend you to follow the terms of the diamond-3 pattern
  - This project is a modified diamond-3 pattern
- Read whole report and modifiers section for more information

09. December 2022:

- Read whole report and modifiers section for more information

SWC Attacks

ID	Title	Relationships	Status
[SWC-136](#)	Unencrypted Private Data On-Chain	[CWE-767: Access to Critical Private Variable via Public Method](#)	**PASSED**
[SWC-135](#)	Code With No Effects	[CWE-1164: Irrelevant Code](#)	**PASSED**
[SWC-134](#)	Message call with hardcoded gas amount	[CWE-655: Improper Initialization](#)	**PASSED**
[SWC-133](#)	Hash Collisions With Multiple Variable Length Arguments	[CWE-294: Authentication Bypass by Capture-replay](#)	**PASSED**
[SWC-132](#)	Unexpected Ether balance	[CWE-667: Improper Locking](#)	**PASSED**
[SWC-131](#)	Presence of unused variables	[CWE-1164: Irrelevant Code](#)	**PASSED**
[SWC-130](#)	Right-To-Left-Override control character (U+202E)	[CWE-451: User Interface (UI) Misrepresentation of Critical Information](#)	**PASSED**
[SWC-129](#)	Typographical Error	[CWE-480: Use of Incorrect Operator](#)	**PASSED**
[SWC-128](#)	DoS With Block Gas Limit	[CWE-400: Uncontrolled Resource Consumption](#)	**PASSED**

SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	PASSED
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	PASSED
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	PASSED
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	PASSED

SWC-116	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	**PASSED**
SWC-115	Authorization through tx.origin	CWE-477: Use of Obsolete Function	**PASSED**
SWC-114	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	**PASSED**
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	**PASSED**
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	**PASSED**
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	**PASSED**
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	**PASSED**
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	**PASSED**
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	**PASSED**
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	**PASSED**
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	**PASSED**

SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	PASSED
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	PASSED
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	PASSED
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	PASSED
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	PASSED
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED