Smart Contract Security Audit V1

HEXCEL Token Smart Contract

3/11/2022

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Platform**: Ethereum
- **Contract Address**: 0x84903595953a8de930f13e1a9dab93d2a835f632
- **Website**:
- **Twitter**:
- **Telegram**:
- **Code Source:** https://etherscan.io/address/0x84903595953a8de930f13e1a9dab93d2a835f632#code

Contracts address deployed to test net (Ethereum)

HEXCEL Token smart contracts on Ethereum test-net by the auditor to test every function .

https://goerli.etherscan.io/address/0xaef9eedf60e6da981e52a2c2782375ca5ac97b9f

Token Information:

Name	hexcel
Symbol	HEXCEL
Total supply	5,555,555
Decimals	18
Router	0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D
Reward Token	0x2b591e99afE9f32eAA6214f7B7629768c40Eeb39
Buy Back Fee	0%
Liquidity Fee	1%
Burn Fee	1%
Reward Fee	3%
Max Wallet Limit	5555555
Max Transaction Amount	5555555
Buy Back Upper Limit Amount	1
Fee Wallet	0x0000000000000000000000000000000000000000
Fee Wallet Charity	0x0000000000000000000000000000000000000000
Minimum Balance for Dividend	555
Number of token to sell to add liquidity	5555.555
PCS V2 Pair	0x4eaF68CA551b48db96425FE3D2a390D07d3151C9
PCS V2 Router	0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well Secured**.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 3 low, 0 very low-level issues and 1 note in all solidity files of the contract

The files:

HEXCEL.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
HEXCEL.sol	**963c14bea10d8b589d717d5b739fe8eed02fa8a839d737b665bacc4d44fe189d**	0x84903595953a8de930f13e1a9dab93d2a835f632

- Contract: HEXCEL
- Inherit: Context, IERC20, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
_name	✓	Read / public	**Passed**
_symbol	✓	Read / public	**Passed**
decimals	✓	Read / public	**Passed**
totalSupply	✓	Read / public	**Passed**
allowance	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
_buybackFee	✓	Read / public	**Passed**
_burnFee	✓	Read / public	**Passed**
_liquidityFee	✓	Read / public	**Passed**
_maxTxAmount	✓	Read / public	**Passed**
_maxWalletAmount	✓	Read / public	**Passed**
_rewardFee	✓	Read / public	**Passed**

_taxFee	✓	Read / public	**Passed**
_tDividendTotal	✓	Read / public	**Passed**
pcsV2Router	✓	Read / public	**Passed**
_tTotal	✓	Read / public	**Passed**
_walletCharityFee	✓	Read / public	**Passed**
_walletFee	✓	Read / public	**Passed**
accumulativeDividendOf	✓	Read / public	**Passed**
buyBackUpperLimitAmount	✓	Read / public	**Passed**
claimWait	✓	Read / public	**Passed**
dividendOf	✓	Read / public	**Passed**
excludedFromDividends	✓	Read / public	**Passed**
feeWalletCharity	✓	Read / public	**Passed**
feeWallet	✓	Read / public	**Passed**
gasForProcessing	✓	Read / public	**Passed**
getAccountDividendsInfo	✓	Read / public	**Passed**
getAccountDividendsInfoAtIndex	✓	Read / public	**Passed**
getLastProcessedIndex	✓	Read / public	**Passed**
getNumberOfDividendTokenHolders	✓	Read / public	**Passed**
geUnlockTime	✓	Read / public	**Passed**
isExcludedFromReward	✓	Read / public	**Passed**
isExcludedFromFee	✓	Read / public	**Passed**
lastClaimTimes	✓	Read / public	**Passed**
lastProcessedIndex	✓	Read / public	**Passed**
maxBuybackFee	✓	Read / public	**Passed**
maxBurnFee	✓	Read / public	**Passed**
maxLiqFee	✓	Read / public	**Passed**
maxWalletFee	✓	Read / public	**Passed**
maxTaxFee	✓	Read / public	**Passed**

minimumTokenBalanceForDividends	✓	Read / public	**Passed**
minMxTxPercentage	✓	Read / public	**Passed**
minMxWalletPercentage	✓	Read / public	**Passed**
name	✓	Read / public	**Passed**
owner	✓	Read / public	**Passed**
numTokensSellToAddToLiquidity	✓	Read / public	**Passed**
reflectionFromToken	✓	Read / public	**Passed**
pcsV2Pair	✓	Read / public	**Passed**
router	✓	Read / public	**Passed**
rewardToken	✓	Read / public	**Passed**
swapAndLiquifyEnabled	✓	Read / public	**Passed**
tokenFromReflection	✓	Read / public	**Passed**
totalDividendsDistributed	✓	Read / public	**Passed**
totalFees	✓	Read / public	**Passed**
withdrawableDividendOf	✓	Read / public	**Passed**
withdrawnDividendOf	✓	Read / public	**Passed**
approve	✓	Write / public	**Passed**
transferFrom	✓	Write / public	**Passed**
transfer	✓	Write / public	**Passed**
lock	✓	Write / public	**Passed**
unLock	✓	Write / public	**Passed**
claim	✓	Write /public	**Passed**
deliver	✓	Write / public	**Passed**
excludeFromDividends	✓	Write / public	**Passed**
excludeFromReward	✓	Write / public	**Passed**
excludeFromfee	✓	Write / public	**Passed**
includeFromReward	✓	Write / public	**Passed**
includeFromfee	✓	Write / public	**Passed**

increaseAllowance	✓	Write / public	**Passed**
decreaseAllowance	✓	Write / public	**Passed**
processDividendTracker	✓	Write / public	**Passed**
process	✓	Write / public	**Passed**
recoverBEP20	✓	Write / public	**Passed**
renounceOwnership	✓	Write / public	**Passed**
setBuybackUpperLimit	✓	Write / public	**Passed**
setFeeWalletCharity	✓	Write / public	**Passed**
setFeeWallet	✓	Write / public	**Passed**
setMinimumTokenBalanceForDividends	✓	Write / public	**Passed**
setSwapAndLiquifyEnabled	✓	Write / public	**Passed**
setWalletCharityFeeTokenType	✓	Write / public	**Passed**
setWalletFeeTokenType	✓	Write / public	**Passed**
transferOwnership	✓	Write / public	**Passed**
updateClaimWait	✓	Write / public	**Passed**
updateGasForProcessing	✓	Write / public	**Passed**
updatePcsV2Router	✓	Write / public	**Passed**
withdrawDividend	✓	Write / public	**Passed**

Issues Checking Status

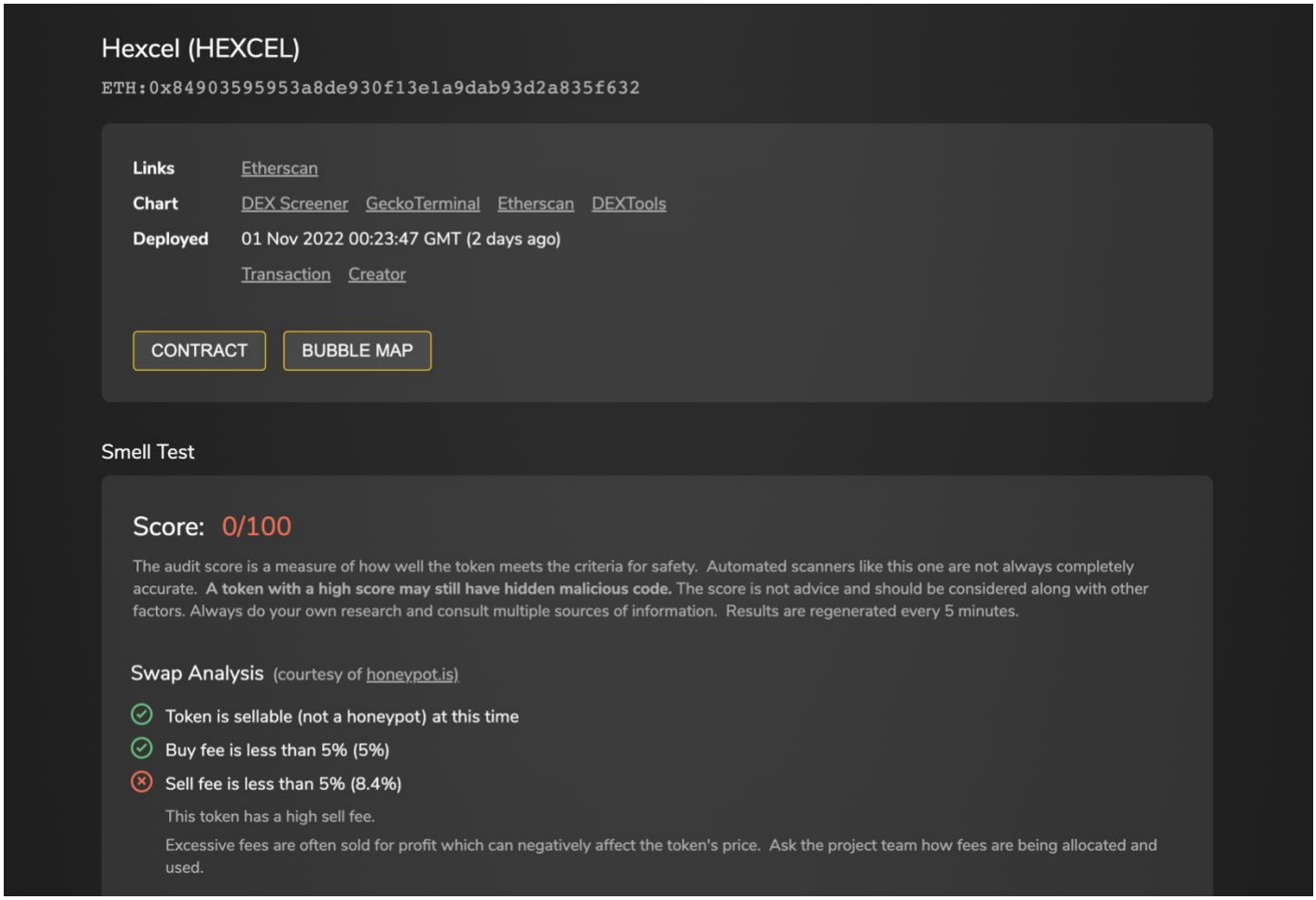
No.	Issue Description	Checking Status
1	Compiler warnings.	**Passed**
2	Race conditions and Reentrancy. Cross-function race conditions.	**Passed**
3	Possible delays in data delivery.	**Passed**
4	Oracle calls.	**Passed**
5	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed with notes**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed with notes**
10	Methods execution permissions.	**Passed**
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	**Passed**
12	The impact of the exchange rate on the logic.	**Passed**
13	Private user data leaks.	**Passed**
14	Malicious Event log.	**Passed**
15	Scoping and Declarations.	**Passed**
16	Uninitialized storage pointers.	**Passed**
17	Arithmetic accuracy.	**Passed**

Severity Definitions

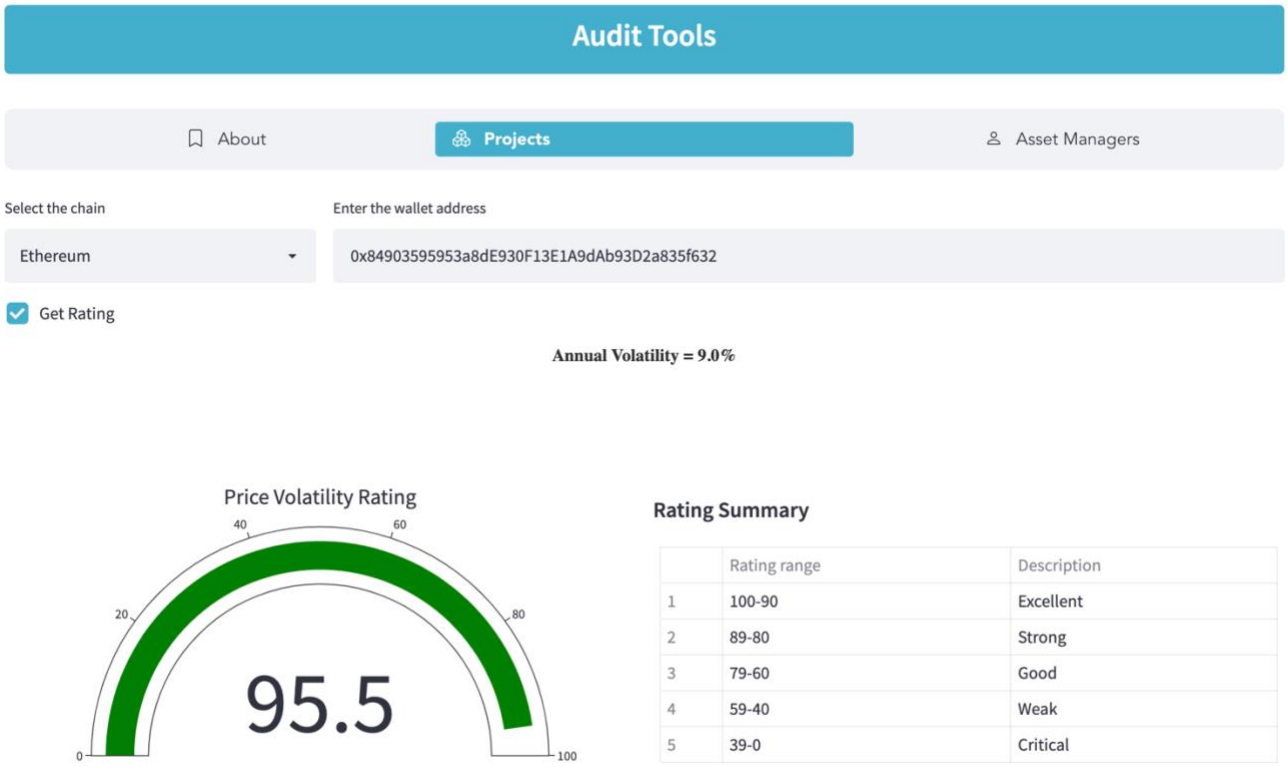
Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

Important Note:

SaferICO certifies that the taxes defined in this audit report are correct. Token Sniffer taxes are wrong. We tested the taxes on Testnet and found taxes written in this report are correct.
Total taxes = 5% (both buy and sell) and it can't be change by the owner.



Price Volatility Rating (Beta):



	Rating range	Description
1	100-90	Excellent
2	89-80	Strong
3	79-60	Good
4	59-40	Weak
5	39-0	Critical

Audit Findings

Critical:

No Critical severity vulnerabilities were found.

High:

No High severity vulnerabilities were found.

Medium:

No Medium severity vulnerabilities were found.

Low:

#Contract code size exceeds 24576 bytes

Description

Contract implementation is too large in size to be deployed on main net. Ethereum with its spurious dragon release limited the size of the contracts deployable on main net to 24576 bytes.
The size of the contract HEXCEL.sol goes way above this value.
You can read more here:
https://github.com/ethereum/EIPs/issues/170

Remediation

Define and use libraries for pure and view functions e.g. We can create a library which contains all the mathematical operations.

Status: Closed. The team used to enable optimization at 200 to avoid this issue.

#Pragam version not fixed

Description

It is a good practice to lock the solidity version for a live deployment (use 0.8.17 instead of ^0.8.15). contracts should be deployed with the same compiler version and flags that they have been tested the most with. Locking the pragma helps ensure that contracts do not accidentally get deployed using, for example, the latest compiler which may have higher risks of undiscovered bugs. Contracts may also be deployed by others and the pragma indicates the compiler version intended by the original authors.

Remediation

Remove the ^ sign to lock the pragma version.

Status: Acknowledged

#Use of block.timestamp for comparisons

Description

The value of block.timestamp can be manipulated by the miner.
And conditions with strict equality is difficult to achieve -
block.timestamp

Remediation
Avoid use of block.timestamp

Status: Acknowledged

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Naming Conventions

Description
The contract follows a consistent naming convention where we are private variables with leading"_" and public variables without it. But we have missed to comply to the condition for certain variable names "__ burnFee " which is public.

Remediation
Remove "_" from external variable names and add it to private variable names.

Status: Acknowledged

Automatic Testing

1- Check for security

963c14bea10d8b589d717d5b739fe8eed02fa8a839d737b665bacc4d44fe189d

File: HEXCE... | Language: solidity | Size: 62050 bytes | Date: 2022-11-03T12:56:05.409Z

Critical	High	Medium	Low	Note
0	0	0	0	0

2- SOLIDITY STATIC ANALYSIS

SOLIDITY STATIC ANALYSIS

Select all | Autorun | Run

Security

Select Security

- **Transaction origin:** 'tx.origin' used
- **Check-effects-interaction:** Potential reentrancy bugs
- **Inline assembly:** Inline assembly used
- **Block timestamp:** Can be influenced by miners
- **Low level calls:** Should only be used by experienced devs
- **Block hash:** Can be influenced by miners
- **Selfdestruct:** Contracts using destructed contract can be broken

Gas & Economy

Select Gas & Economy

- **Gas costs:** Too high gas requirement of functions
- **This on local calls:** Invocation of local functions via 'this'
- **Delete dynamic array:** Use require/assert to ensure complete deletion
- **For loop over dynamic array:** Iterations depend on dynamic array's size
- **Ether transfer in loop:** Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

ERC

Select ERC

- **ERC20:** 'decimals' should be 'uint8'

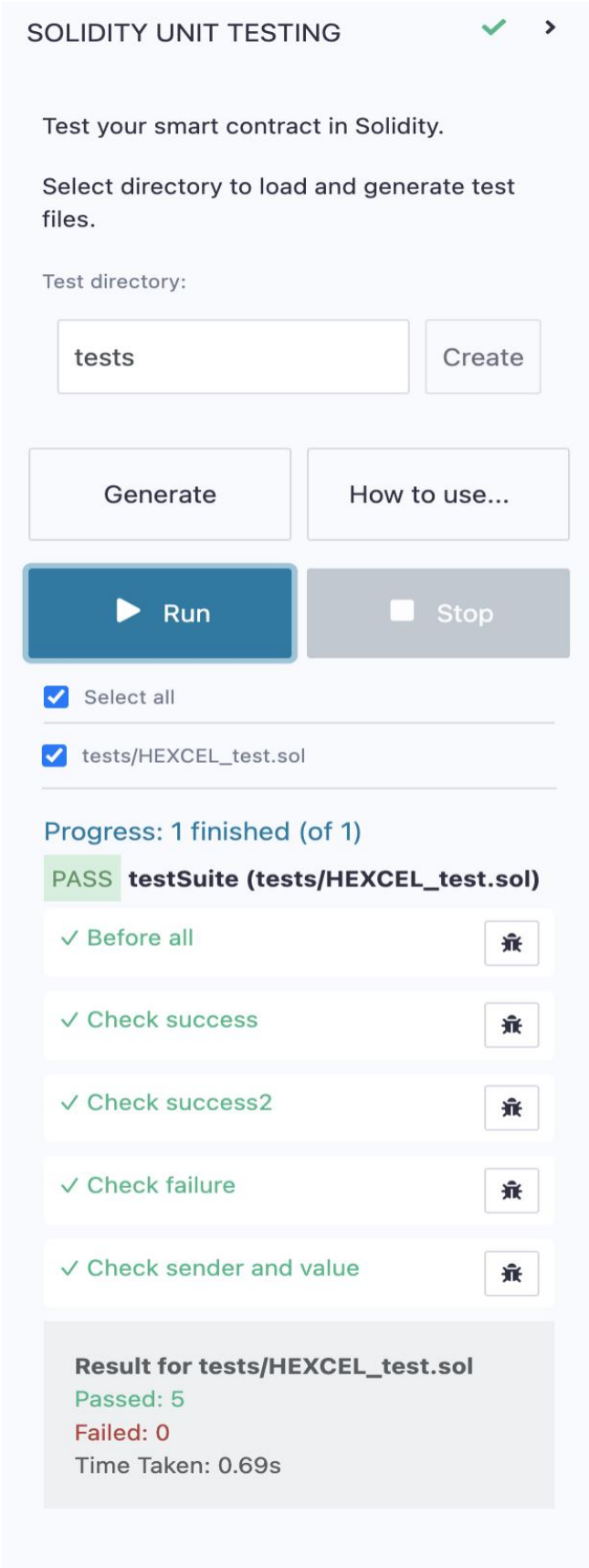
Miscellaneous

Select Miscellaneous

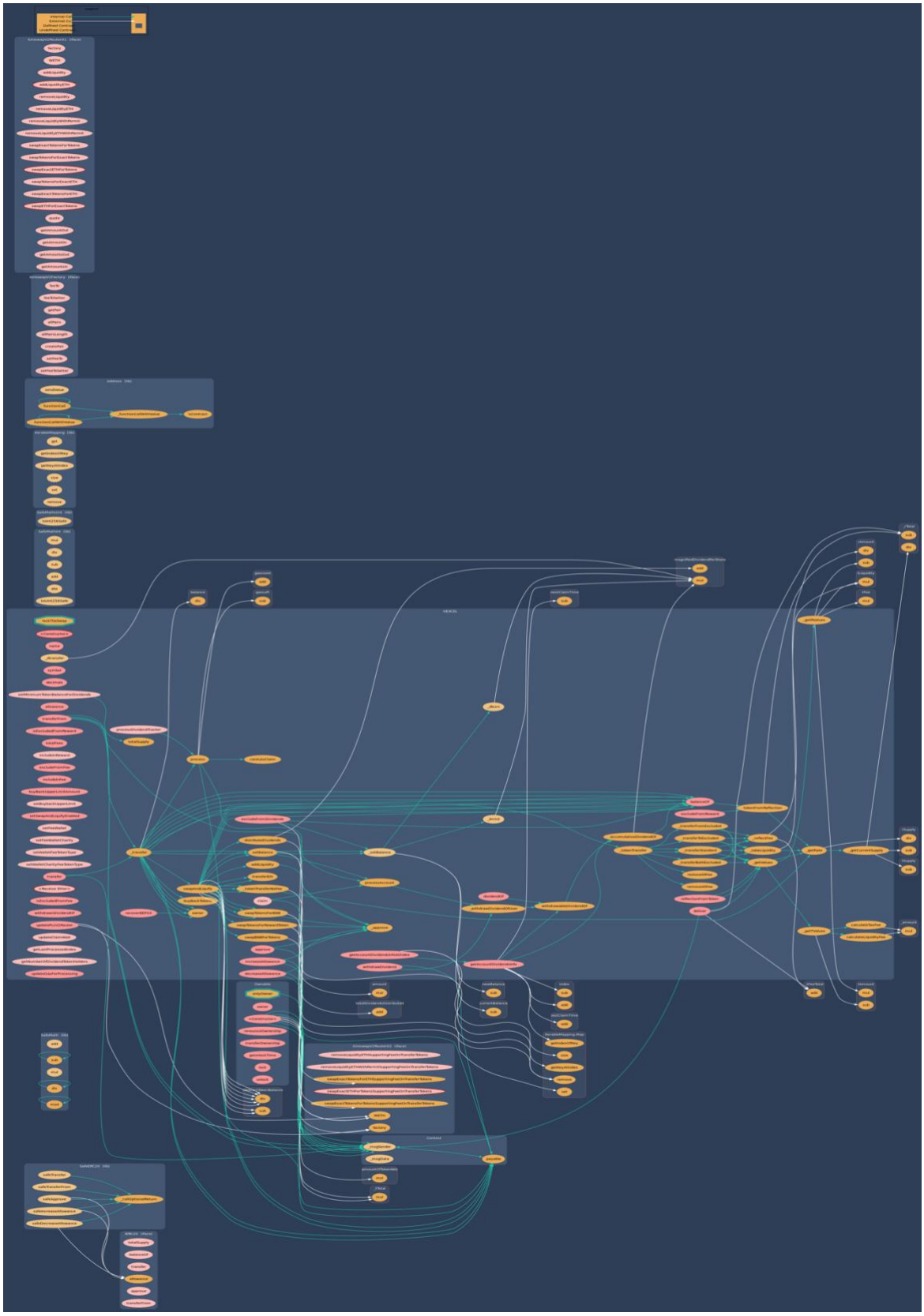
- **Constant/View/Pure functions:** Potentially constant/view/pure functions
- **Similar variable names:** Variable names are too similar
- **No return:** Function with 'returns' not returning
- **Guard conditions:** Ensure appropriate use of require/assert
- **Result not used:** The result of an operation not used
- **String length:** Bytes length != String length
- **Delete from dynamic array:** 'delete' leaves a gap in array
- **Data truncated:** Division on int/uint values truncates the result

3- Inheritance graph

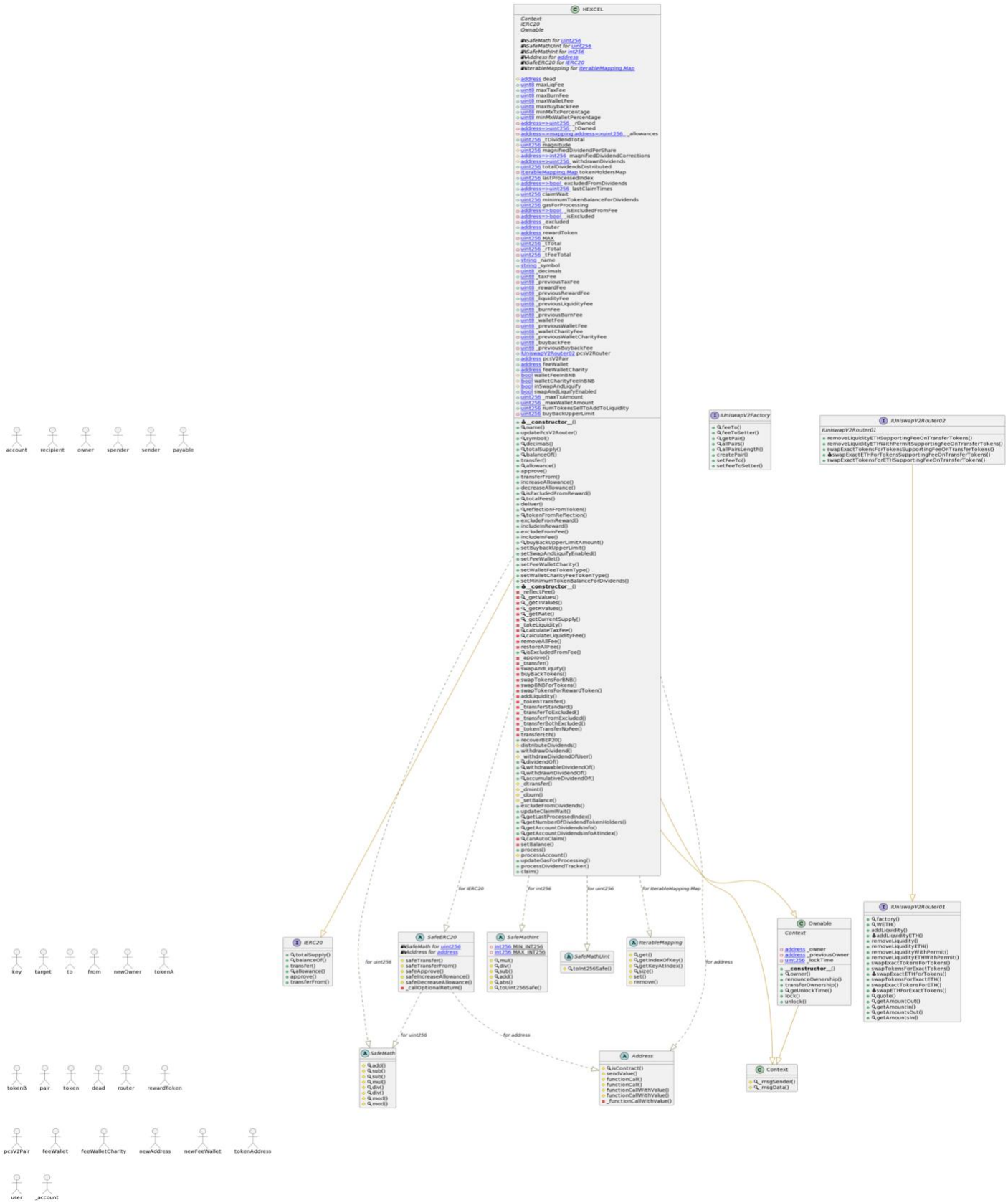
4- SOLIDITY UNIT TESTING



5- Call graph



Unified Modeling Language (UML)



Functions signature

```
Sighash   |   Function Signature
========================
11902160  =>  _getTValues(uint256)
16279055  =>  isContract(address)
39509351  =>  increaseAllowance(address,uint256)
43509138  =>  div(int256,int256)
75128141  =>  calculateTaxFee(uint256)
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
b745d336  =>  div(uint256,uint256,string)
f43f523a  =>  mod(uint256,uint256)
71af23e8  =>  mod(uint256,uint256,string)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
bbe93d91  =>  mul(int256,int256)
adefc37b  =>  sub(int256,int256)
a5f3c23b  =>  add(int256,int256)
1b5ac4b5  =>  abs(int256)
744f7c7d  =>  toUint256Safe(int256)
e823b9bf  =>  toInt256Safe(uint256)
268d8e2e  =>  get(Map,address)
b45dad3d  =>  getIndexOfKey(Map,address)
7596720f  =>  getKeyAtIndex(Map,uint256)
b1b533f3  =>  size(Map)
6b06f325  =>  set(Map,address,uint256)
0eac8729  =>  remove(Map,address)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
36455e42  =>  _functionCallWithValue(address,bytes,uint256,string)
d0c407e1  =>  safeTransfer(IERC20,address,uint256)
5beae096  =>  safeTransferFrom(IERC20,address,address,uint256)
d6dcec8d  =>  safeApprove(IERC20,address,uint256)
390cc046  =>  safeIncreaseAllowance(IERC20,address,uint256)
5164ffed  =>  safeDecreaseAllowance(IERC20,address,uint256)
becc5a20  =>  _callOptionalReturn(IERC20,bytes)
8da5cb5b  =>  owner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
b6c52324  =>  geUnlockTime()
dd467064  =>  lock(uint256)
a69df4b5  =>  unlock()
017e7e58  =>  feeTo()
094b7415  =>  feeToSetter()
e6a43905  =>  getPair(address,address)
```

```
1e3dd18b  =>  allPairs(uint256)
574f2ba3  =>  allPairsLength()
c9c65396  =>  createPair(address,address)
f46901ed  =>  setFeeTo(address)
a2e74af6  =>  setFeeToSetter(address)
c45a0155  =>  factory()
ad5c4648  =>  WETH()
e8e33700  =>
addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256)
f305d719  =>  addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
baa2abde  =>
removeLiquidity(address,address,uint256,uint256,uint256,address,uint256)
02751cec  =>  removeLiquidityETH(address,uint256,uint256,uint256,address,uint256)
2195995c  =>
removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,b
ool,uint8,bytes32,bytes32)
ded9382a  =>
removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,u
int8,bytes32,bytes32)
38ed1739  =>  swapExactTokensForTokens(uint256,uint256,address[],address,uint256)
8803dbee  =>  swapTokensForExactTokens(uint256,uint256,address[],address,uint256)
7ff36ab5  =>  swapExactETHForTokens(uint256,address[],address,uint256)
4a25d94a  =>  swapTokensForExactETH(uint256,uint256,address[],address,uint256)
18cbafe5  =>  swapExactTokensForETH(uint256,uint256,address[],address,uint256)
fb3bdb41  =>  swapETHForExactTokens(uint256,address[],address,uint256)
ad615dec  =>  quote(uint256,uint256,uint256)
054d50d4  =>  getAmountOut(uint256,uint256,uint256)
85f8c259  =>  getAmountIn(uint256,uint256,uint256)
d06ca61f  =>  getAmountsOut(uint256,address[])
1f00ca74  =>  getAmountsIn(uint256,address[])
af2979eb  =>
removeLiquidityETHSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,add
ress,uint256)
5b0d5984  =>
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(address,uint256,uint256,u
int256,address,uint256,bool,uint8,bytes32,bytes32)
5c11d795  =>
swapExactTokensForTokensSupportingFeeOnTransferTokens(uint256,uint256,address[],add
ress,uint256)
b6f9de95  =>
swapExactETHForTokensSupportingFeeOnTransferTokens(uint256,address[],address,uint25
6)
791ac947  =>
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],addres
s,uint256)
06fdde03  =>  name()
caebb843  =>  updatePcsV2Router(address)
95d89b41  =>  symbol()
313ce567  =>  decimals()
a457c2d7  =>  decreaseAllowance(address,uint256)
88f82020  =>  isExcludedFromReward(address)
13114a9d  =>  totalFees()
3bd5d173  =>  deliver(uint256)
4549b039  =>  reflectionFromToken(uint256,bool)
2d838119  =>  tokenFromReflection(uint256)
52390c02  =>  excludeFromReward(address)
3685d419  =>  includeInReward(address)
437823ec  =>  excludeFromFee(address)
ea2f0b37  =>  includeInFee(address)
```

```
bdc653ef  =>  buyBackUpperLimitAmount()
82d2a4bb  =>  setBuybackUpperLimit(uint256)
c49b9a80  =>  setSwapAndLiquifyEnabled(bool)
90d49b9d  =>  setFeeWallet(address)
a79771bb  =>  setFeeWalletCharity(address)
7bd56c8c  =>  setWalletFeeTokenType(bool)
a5f35177  =>  setWalletCharityFeeTokenType(bool)
5ebf4db9  =>  setMinimumTokenBalanceForDividends(uint256)
184d894e  =>  _reflectFee(uint256,uint256)
d4780e36  =>  _getValues(uint256)
1d5671e4  =>  _getRValues(uint256,uint256,uint256,uint256)
94e10784  =>  _getRate()
97a9d560  =>  _getCurrentSupply()
c432df5e  =>  _takeLiquidity(uint256)
cc126a23  =>  calculateLiquidityFee(uint256)
301370af  =>  removeAllFee()
e7e3e3a7  =>  restoreAllFee()
5342acb4  =>  isExcludedFromFee(address)
104e81ff  =>  _approve(address,address,uint256)
30e0789e  =>  _transfer(address,address,uint256)
173865ad  =>  swapAndLiquify(uint256)
fc155d1d  =>  buyBackTokens(uint256)
56c3726b  =>  swapTokensForBNB(uint256)
cf36bec9  =>  swapBNBForTokens(uint256)
589ad64e  =>  swapTokensForRewardToken(uint256)
9cd441da  =>  addLiquidity(uint256,uint256)
b09bbc79  =>  _tokenTransfer(address,address,uint256,bool)
2852df65  =>  _transferStandard(address,address,uint256)
16f1cc83  =>  _transferToExcluded(address,address,uint256)
c7d9be66  =>  _transferFromExcluded(address,address,uint256)
6ff6cdf4  =>  _transferBothExcluded(address,address,uint256)
de621616  =>  _tokenTransferNoFee(address,address,uint256)
e9bb84c2  =>  transferEth(address,uint256)
efa08806  =>  recoverBEP20(address,uint256)
3243c791  =>  distributeDividends(uint256)
6a474002  =>  withdrawDividend()
373de4aa  =>  _withdrawDividendOfUser(address)
91b89fba  =>  dividendOf(address)
a8b9d240  =>  withdrawableDividendOf(address)
aafd847a  =>  withdrawnDividendOf(address)
27ce0147  =>  accumulativeDividendOf(address)
721bf495  =>  _dtransfer(address,address,uint256)
19f93152  =>  _dmint(address,uint256)
8f7e2e92  =>  _dburn(address,uint256)
7eae6759  =>  _setBalance(address,uint256,uint256)
31e79db0  =>  excludeFromDividends(address)
e98030c7  =>  updateClaimWait(uint256)
e7841ec0  =>  getLastProcessedIndex()
64b0f653  =>  getNumberOfDividendTokenHolders()
ad56c13c  =>  getAccountDividendsInfo(address)
f27fd254  =>  getAccountDividendsInfoAtIndex(uint256)
77fdb837  =>  canAutoClaim(uint256)
e8a6a289  =>  setBalance(address,uint256,uint256)
ffb2c479  =>  process(uint256)
bc4c4b37  =>  processAccount(address,bool)
871c128d  =>  updateGasForProcessing(uint256)
700bb191  =>  processDividendTracker(uint256)
4e71d92d  =>  claim()
```

Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/HEXCEL.sol |
d36f355edc1b4bcde1c069c6058765e436835b2c |

 Contracts Description Table

|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:--------
-------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗ |   |NO❗ |
| └ | balanceOf | External ❗ |   |NO❗ |
| └ | transfer | External ❗ | 🛑  |NO❗ |
| └ | allowance | External ❗ |   |NO❗ |
| └ | approve | External ❗ | 🛑  |NO❗ |
| └ | transferFrom | External ❗ | 🛑  |NO❗ |
||||||
| **SafeMath** | Library |  |||
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **SafeMathInt** | Library |  |||
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | add | Internal 🔒 |   | |
| └ | abs | Internal 🔒 |   | |
| └ | toUint256Safe | Internal 🔒 |   | |
||||||
| **SafeMathUint** | Library |  |||
| └ | toInt256Safe | Internal 🔒 |   | |
||||||
| **IterableMapping** | Library |  |||
| └ | get | Internal 🔒 |   | |
| └ | getIndexOfKey | Internal 🔒 |   | |
| └ | getKeyAtIndex | Internal 🔒 |   | |
| └ | size | Internal 🔒 |   | |
| └ | set | Internal 🔒 | 🛑  | |
```

└	remove	Internal 🔒	🛑	
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
SafeERC20	Library			
└	safeTransfer	Internal 🔒	🛑	
└	safeTransferFrom	Internal 🔒	🛑	
└	safeApprove	Internal 🔒	🛑	
└	safeIncreaseAllowance	Internal 🔒	🛑	
└	safeDecreaseAllowance	Internal 🔒	🛑	
└	_callOptionalReturn	Private 🔐	🛑	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	geUnlockTime	Public ❗		NO❗
└	lock	Public ❗	🛑	onlyOwner
└	unlock	Public ❗	🛑	NO❗
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2Router01	Interface			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💶	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💶	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗
└	swapETHForExactTokens	External ❗	💶	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗

```
|||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🔘 |NO❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🔘
|NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🔘 |NO❗
|
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💰 |NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🔘 |NO❗ |
|||||
| **HEXCEL** | Implementation | Context, IERC20, Ownable |||
| └ | <Constructor> | Public ❗ | 💰 |NO❗ |
| └ | name | Public ❗ | |NO❗ |
| └ | updatePcsV2Router | Public ❗ | 🔘 | onlyOwner |
| └ | symbol | Public ❗ | |NO❗ |
| └ | decimals | Public ❗ | |NO❗ |
| └ | totalSupply | Public ❗ | |NO❗ |
| └ | balanceOf | Public ❗ | |NO❗ |
| └ | transfer | Public ❗ | 🔘 |NO❗ |
| └ | allowance | Public ❗ | |NO❗ |
| └ | approve | Public ❗ | 🔘 |NO❗ |
| └ | transferFrom | Public ❗ | 🔘 |NO❗ |
| └ | increaseAllowance | Public ❗ | 🔘 |NO❗ |
| └ | decreaseAllowance | Public ❗ | 🔘 |NO❗ |
| └ | isExcludedFromReward | Public ❗ | |NO❗ |
| └ | totalFees | Public ❗ | |NO❗ |
| └ | deliver | Public ❗ | 🔘 |NO❗ |
| └ | reflectionFromToken | Public ❗ | |NO❗ |
| └ | tokenFromReflection | Public ❗ | |NO❗ |
| └ | excludeFromReward | Public ❗ | 🔘 | onlyOwner |
| └ | includeInReward | External ❗ | 🔘 | onlyOwner |
| └ | excludeFromFee | Public ❗ | 🔘 | onlyOwner |
| └ | includeInFee | Public ❗ | 🔘 | onlyOwner |
| └ | buyBackUpperLimitAmount | Public ❗ | |NO❗ |
| └ | setBuybackUpperLimit | External ❗ | 🔘 | onlyOwner |
| └ | setSwapAndLiquifyEnabled | Public ❗ | 🔘 | onlyOwner |
| └ | setFeeWallet | External ❗ | 🔘 | onlyOwner |
| └ | setFeeWalletCharity | External ❗ | 🔘 | onlyOwner |
| └ | setWalletFeeTokenType | External ❗ | 🔘 | onlyOwner |
| └ | setWalletCharityFeeTokenType | External ❗ | 🔘 | onlyOwner |
| └ | setMinimumTokenBalanceForDividends | External ❗ | 🔘 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💰 |NO❗ |
| └ | _reflectFee | Private 🔐 | 🔘 | |
| └ | _getValues | Private 🔐 | | |
| └ | _getTValues | Private 🔐 | | |
| └ | _getRValues | Private 🔐 | | |
| └ | _getRate | Private 🔐 | | |
| └ | _getCurrentSupply | Private 🔐 | | |
| └ | _takeLiquidity | Private 🔐 | 🔘 | |
| └ | calculateTaxFee | Private 🔐 | | |
| └ | calculateLiquidityFee | Private 🔐 | | |
| └ | removeAllFee | Private 🔐 | 🔘 | |
| └ | restoreAllFee | Private 🔐 | 🔘 | |
| └ | isExcludedFromFee | Public ❗ | |NO❗ |
| └ | _approve | Private 🔐 | 🔘 | |
| └ | _transfer | Private 🔐 | 🔘 | |
| └ | swapAndLiquify | Private 🔐 | 🔘 | lockTheSwap |
| └ | buyBackTokens | Private 🔐 | 🔘 | lockTheSwap |
```

└	swapTokensForBNB	Private 🔐	🛑	
└	swapBNBForTokens	Private 🔐	🛑	
└	swapTokensForRewardToken	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	_tokenTransferNoFee	Private 🔐	🛑	
└	transferEth	Private 🔐	🛑	
└	recoverBEP20	Public ❗	🛑	onlyOwner
└	distributeDividends	Internal 🔒	🛑	
└	withdrawDividend	Public ❗	🛑	NO❗
└	_withdrawDividendOfUser	Internal 🔒	🛑	
└	dividendOf	Public ❗		NO❗
└	withdrawableDividendOf	Public ❗		NO❗
└	withdrawnDividendOf	Public ❗		NO❗
└	accumulativeDividendOf	Public ❗		NO❗
└	_dtransfer	Internal 🔒	🛑	
└	_dmint	Internal 🔒	🛑	
└	_dburn	Internal 🔒	🛑	
└	_setBalance	Internal 🔒	🛑	
└	excludeFromDividends	Public ❗	🛑	onlyOwner
└	updateClaimWait	External ❗	🛑	onlyOwner
└	getLastProcessedIndex	External ❗		NO❗
└	getNumberOfDividendTokenHolders	External ❗		NO❗
└	getAccountDividendsInfo	Public ❗		NO❗
└	getAccountDividendsInfoAtIndex	Public ❗		NO❗
└	canAutoClaim	Private 🔐		
└	setBalance	Private 🔐	🛑	
└	process	Public ❗	🛑	NO❗
└	processAccount	Internal 🔒	🛑	
└	updateGasForProcessing	Public ❗	🛑	onlyOwner
└	processDividendTracker	External ❗	🛑	NO❗
└	claim	External ❗	🛑	NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💶	Function is payable

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “Well Secured”.

✓ No mint function.
✓ No volatile code.
✓ No high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.