



**SOLID**Proof  
*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC**

MADE IN GERMANY

# Audit

**Security Assessment**  
**22. October, 2021**



MilkywayEX

Disclaimer	3
Description	5
Project Engagement	5
Logo	5
Contract Link	5
Methodology	7
Used Code from other Frameworks/Smart Contracts (direct imports)	8
Tested Contract Files	9
Source Lines	11
Risk Level	11
Capabilities	12
Scope of Work	14
Inheritance Graph	14
Verify Claims	15
OnlyOwner functions	21
CallGraph	22
Source Units in Scope	23
Critical issues	25
High issues	25
Medium issues	25
Low issues	25
Informational issues	26
Audit Comments	26
SWC Attacks	27

# 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 Uniswap, 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	22. October 2021	<ul style="list-style-type: none"><li>• Layout project</li><li>• Automated- /Manual-Security Testing</li><li>• Summary</li></ul>
1.1	22. October 2022	<ul style="list-style-type: none"><li>• Fixed medium issue</li></ul>

## **Network**

Binance Smart Chain (BEP20)

## **Website**

<https://milkywayex.finance/>

## **Medium**

<https://milkywayex.medium.com/>



## Description

**MilkywayEx** is your one-stop decentralized trading platform on the Binance Smart Chain network.

We combine DEX services with DeFi rewarding to offer leveraged trading.

## Project Engagement

During the 18th of October 2021, **MilkyWayEx 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

### v1.0

<https://bscscan.com/address/0x39d248e2979910351cc5892fdf420df0fb5f0f68#code>

### v1.1

<https://bscscan.com/address/0xa7002fccc20f10a38c579a8d2d14d34f2a9a02a5#code>

# 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)
<b>Critical</b>	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.
<b>High</b>	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 as possible.
<b>Medium</b>	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.
<b>Low</b>	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.
<b>Informational</b>	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.

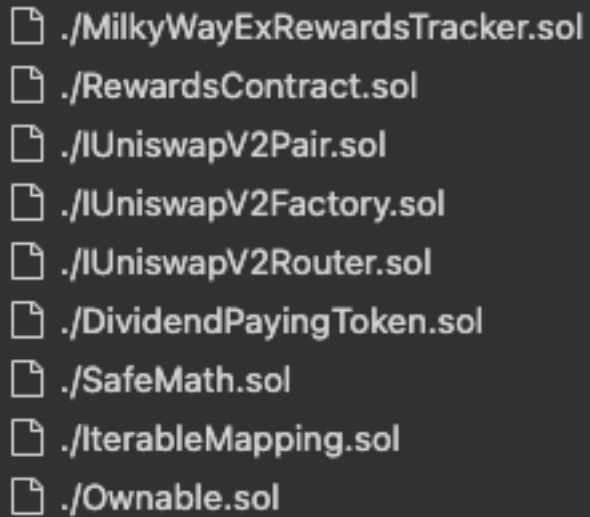
## **Methodology**

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:



- ./MilkyWayExRewardsTracker.sol
- ./RewardsContract.sol
- ./IUniswapV2Pair.sol
- ./IUniswapV2Factory.sol
- ./IUniswapV2Router.sol
- ./DividendPayingToken.sol
- ./SafeMath.sol
- ./IterableMapping.sol
- ./Ownable.sol



## 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.*

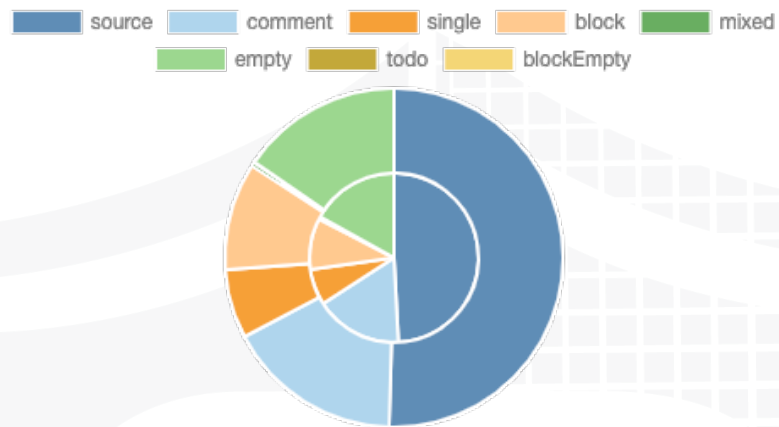
### v1.0

File Name	SHA-1 Hash
contracts/DividendPayingToken.sol	17d53cb94689058b901400a6784c037e3af36bb0
contracts/MilkyWayExRewardsTracker.sol	04aca27460e461b47d5ac6216a296332e9da8df1
contracts/IUniswapV2Pair.sol	a5f9c99358e8488f73b373148f031d4b11e935e7
contracts/RewardsContract.sol	ad4a6d0940b4b31c01f595457e92252a81104089
contracts/SafeMathUint.sol	8b0e5a5ecff9df63b44f5a3b2e382772e8e22b4
contracts/MilkyWayExToken.sol	58b2844159322edc4f6f82ad169b535d9119a128
contracts/Context.sol	e0c372022e49b1d2923932f29603ae4e0a6cad75
contracts/IUniswapV2Factory.sol	ea3cec8ac8aa96b839615f6c09c45b0bb589c309
contracts/DividendPayingTokenInterface.sol	5d601031c738ef91b52629287badfcef5485bdc0
contracts/IERC20Metadata.sol	4cc38ac1b6e456242c07fffbe9921ddf3de260cb
contracts/SafeMathInt.sol	54330e881076020b806231dde99ff0e13fee9a1e
contracts/SafeMath.sol	cff19c662ce063d1e29d5b197111e7cc454e1a51
contracts/IUniswapV2Router.sol	b5a32b744d4f23c4a7baee28760b0765ae1fe145
contracts/Ownable.sol	b40c1f66bb4745e9ec498ff9af16dfb59f4e7d60
contracts/IterableMapping.sol	2e22f708ad8a140954a27d47906ba5f9978c8c19
contracts/ERC20.sol	248ce26e2fa5b02dd441a83e9348adc4e829aad7
contracts/DividendPayingTokenOptionalInterface.sol	243ea42fd26ba8ec267a8d7a80d809d1ebea2a85
contracts/IERC20.sol	64a882695eb836b8c1065d6af0dacfe9c71d5f4b

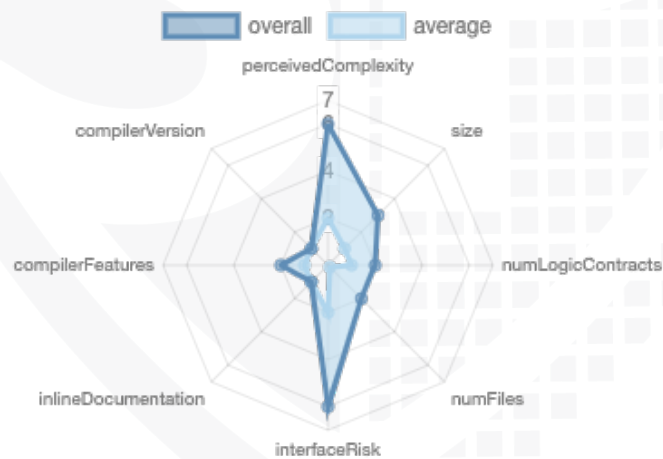
File Name	SHA-1 Hash
contracts/IERC20.sol	64a882695eb836b8c1065d6af0dacfe9c71d5f4b
contracts/IUniswapV2Router.sol	b5a32b744d4f23c4a7baee28760b0765ae1fe145
contracts/SafeMath.sol	cff19c662ce063d1e29d5b197111e7cc454e1a51
contracts/Ownable.sol	b40c1f66bb4745e9ec498ff9af16dfb59f4e7d60
contracts/IterableMapping.sol	2e22f708ad8a140954a27d47906ba5f9978c8c19
contracts/ERC20.sol	248ce26e2fa5b02dd441a83e9348adc4e829aad7
contracts/DividendPayingTokenOptionalInterface.sol	243ea42fd26ba8ec267a8d7a80d809d1e2a2a85
contracts/DividendPayingToken.sol	17d53cb94689058b901400a6784c037e3af36bb0
contracts/MilkyWayExRewardsTracker.sol	04aca27460e461b47d5ac6216a296332e9da8df1
contracts/IUniswapV2Pair.sol	a5f9c99358e8488f73b373148f031d4b11e935e7
contracts/RewardsContract.sol	ad4a6d0940b4b31c01f595457e92252a81104089
contracts/SafeMathUint.sol	8b0e5a5ecff9df63b44f5a3b2e382772e8e22b4
contracts/MilkyWayExToken.sol	8656251bd696adc2d18c514443d2a5b80e3418d9
contracts/Context.sol	e0c372022e49b1d2923932f29603ae4e0a6cad75
contracts/IUniswapV2Factory.sol	ea3cec8ac8aa96b839615f6c09c45b0bb589c309
contracts/DividendPayingTokenInterface.sol	5d601031c738ef91b52629287badfcef5485bdc0
contracts/IERC20Metadata.sol	4cc38ac1b6e456242c07ffbe9921ddf3de260cb
contracts/SafeMathInt.sol	54330e881076020b806231dde99ff0e13fee9a1e

# Metrics

## Source Lines v1.0



## Risk Level v1.0



## Capabilities

### Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	6	4	8	1

### 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	174	8

Version	External	Internal	Private	Pure	View
1.0	115	136	6	24	74

### State Variables

Version	Total	Public
1.0	60	41

### Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	<code>^0.7.6</code>		yes	**** (0 asm blocks)	

Version	Transfers ETH	Low-Level Calls	Delegation Call	Uses Hash Functions	ECRecover	New/ Create/ Create 2
1.0						yes → NewContract:MilkyWayEx RewardsTracker → NewContract:Re wardsContract

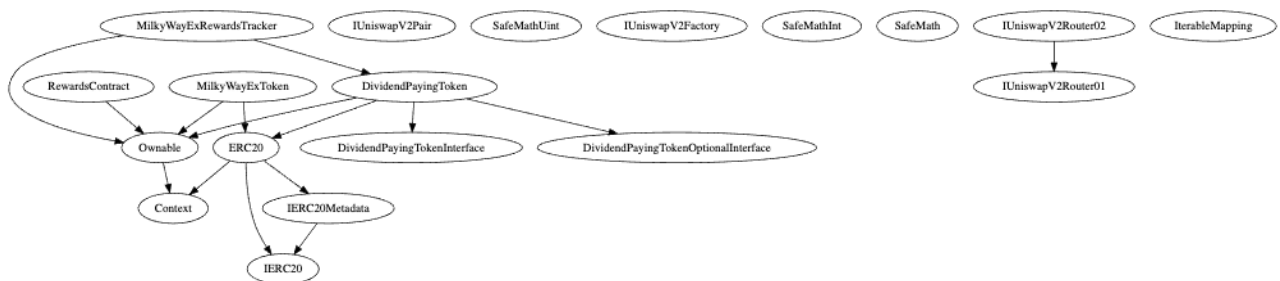
## Scope of Work

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. Correct implementation of Token standard
2. Deployer cannot mint any new tokens
3. Deployer cannot burn or lock user funds
4. Deployer cannot pause the contract
5. Overall checkup (Smart Contract Security)

## Inheritance Graph v1.0



## Verify Claims

### Correct implementation of Token standard

Tested	Verified
✓	✓

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	✓	✓	✓

## Write functions of contract

1. approve	→
2. changeBotWallet	→
3. changeMinimumBalanceToReceiveRewards	→
4. changeRewardsPercentage	→
5. changeRewardsToken	→
6. changeUserClaimTokenPercentage	→
7. changeUserCustomToken	→
8. changeUserRewardsSetup	→
9. claim	→
10. clearHolderRewardsOffset	→
11. decreaseAllowance	→
12. elonSet	→
13. excludeFromDividends	→
14. excludeFromFees	→
15. includeInDividends	→
16. increaseAllowance	→
17. processRewardsTracker	→
18. resetUserClaimTokenPercentage	→
19. resetUserCustomToken	→
20. rewardsAdd	→
21. rewardsRemove	→
22. rewardsSend	→
23. rewardsTime	→
24. transfer	→
25. transferFrom	→
26. transferOwnership	→
27. updateGasForProcessing	→
28. updateHolderRewardsOffset	→
29. updateSingleHolderRewardsOffset	→
30. withdrawETH	→



## Deployer cannot mint any new tokens

Name	Exist	Tested	Verified	File
Deployer cannot mint	✓	✓	✓	Main
Comment	Line: -			

Max / Total Supply: 16.000.000.000



## Deployer cannot burn or lock user funds

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

Comments:

### v1.0

- Deployer can lock user funds
  - If address is added to preventer mapping in RewardsContract.sol
- Deployer can lock dividends if address is added to excludedFromDividends mapping with excludeFromDividends function

## Deployer cannot pause the contract

Name	Exist	Tested	Verified
Deployer cannot pause	✓	✓	✓



## Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

### Legend

Attribute	Symbol
Verified / Checked	✓
Partly Verified	⚠
Unverified / Not checked	✗
Not available	—

## OnlyOwner functions

rewardsAdd

rewardsRemove

excludeFromDividends

includeInDividends

rewardsSend

rewardsTime

excludeFromFees

withdrawETH

elonSet

updateGasForProcessing

updateHolderRewardsOffset

updateSingleHolderRewardsOffset

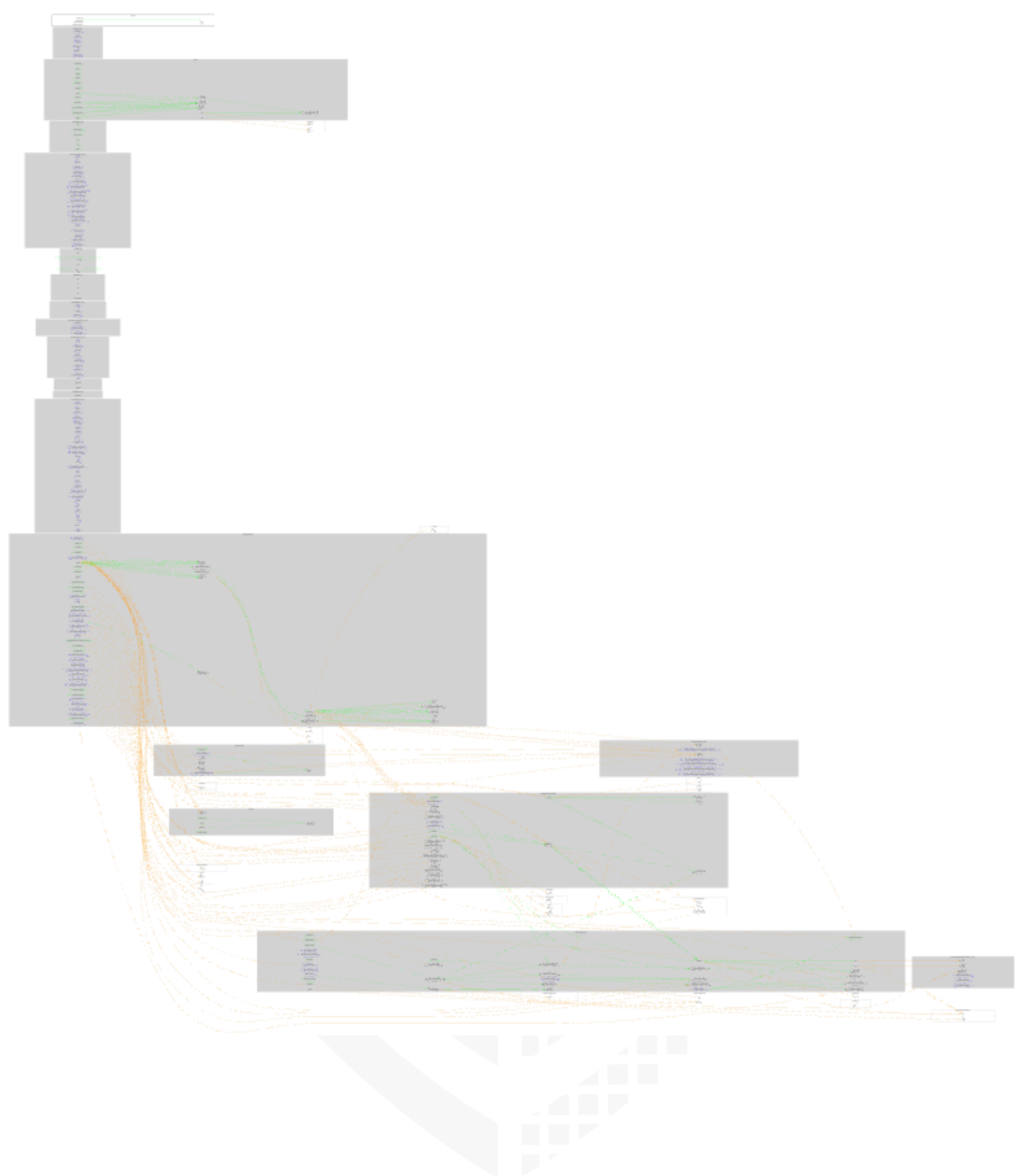
clearHolderRewardsOffset

changeMinimumBalanceToReceiveRewards

changeRewardsPercentage






























changeBotWallet

# CallGraph
































# Source Units in Scope

## v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/DividendPayingToken.sol	1	_____	391	391	239	93	224	
	contracts/MilkyWayExRewardsTracker.sol	1	_____	222	212	158	4	114	_____
	contracts/IUniswapV2Pair.sol	_____	1	55	10	5	1	55	_____
	contracts/RewardsContract.sol	1	_____	68	68	48	3	52	_____
	contracts/SafeMathUint.sol	1	_____	15	15	8	5	3	_____
	contracts/MilkyWayExToken.sol	1	_____	464	450	318	14	320	 
	contracts/Context.sol	1	_____	24	24	10	12	1	_____
	contracts/IUniswapV2Factory.sol	_____	1	20	9	4	1	17	_____
	contracts/DividendPayingTokenInterface.sol	_____	1	40	12	3	20	10	
	contracts/IERC20Metadata.sol	_____	1	19	14	4	6	9	
	contracts/SafeMathInt.sol	1	_____	59	59	29	20	14	_____
	contracts/SafeMath.sol	1	_____	47	47	32	2	8	_____
	contracts/IUniswapV2Router.sol	_____	2	141	8	4	1	64	
	contracts/Ownable.sol	1	_____	47	47	23	14	17	_____
	contracts/IterableMapping.sol	1	_____	65	65	49	2	19	_____
	contracts/ERC20.sol	1	_____	286	270	85	154	80	_____
	contracts/DividendPayingTokenOptionalInterface.sol	_____	1	25	12	3	14	7	_____
	contracts/IERC20.sol	_____	1	82	27	17	57	13	
	<b>Totals</b>	<b>11</b>	<b>8</b>	<b>2070</b>	<b>1740</b>	<b>1039</b>	<b>423</b>	<b>1027</b>	  

## v1.1

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/IERC20.sol	_____	1	82	27	17	57	13	
	contracts/IUniswapV2Router.sol	_____	2	141	8	4	1	64	
	contracts/SafeMath.sol	1	_____	47	47	32	2	8	_____
	contracts/Ownable.sol	1	_____	47	47	23	14	17	_____
	contracts/IterableMapping.sol	1	_____	65	65	49	2	19	_____
	contracts/ERC20.sol	1	_____	286	270	85	154	80	_____
	contracts/DividendPayingTokenOptionalInterface.sol	_____	1	25	12	3	14	7	_____
	contracts/DividendPayingToken.sol	1	_____	391	391	239	93	224	
	contracts/MilkyWayExRewardsTracker.sol	1	_____	222	212	158	4	114	_____
	contracts/IUniswapV2Pair.sol	_____	1	55	10	5	1	55	_____
	contracts/RewardsContract.sol	1	_____	68	68	48	3	52	_____
	contracts/SafeMathUint.sol	1	_____	15	15	8	5	3	_____
	contracts/MilkyWayExToken.sol	1	_____	465	451	319	14	321	 
	contracts/Context.sol	1	_____	24	24	10	12	1	_____
	contracts/IUniswapV2Factory.sol	_____	1	20	9	4	1	17	_____
	contracts/DividendPayingTokenInterface.sol	_____	1	40	12	3	20	10	
	contracts/IERC20Metadata.sol	_____	1	19	14	4	6	9	
	contracts/SafeMathInt.sol	1	_____	59	59	29	20	14	_____
	<b>Totals</b>	<b>11</b>	<b>8</b>	<b>2071</b>	<b>1741</b>	<b>1040</b>	<b>423</b>	<b>1028</b>	  

## Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
nSLOC	normalized 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

## AUDIT PASSED

### Critical issues

- no critical issues found -

### High issues

- no high issues found -

### Medium issues

Issue	File	Type	Line	Description
#1	MilkyWayExToken	Low level calls (low-level-calls)	376	Check the call success. Ensure that the return value of a low-level call is checked or logged

FIXED

### Low issues

Issue	File	Type	Line	Description
#1	MilkyWayExToken	A floating pragma is set	3	The current pragma Solidity directive is „^0.7.6“.
#2	MilkyWayExToken	Missing Zero Address Validation (missing-zero-check)	164	Check that the address is not zero
#3	DividendPayingToken	Missing Zero Address Validation (missing-zero-check)	196	Check that the address is not zero
#4	DividendPayingToken	Tautology or contradiction (tautology)	156, 165	Fix the incorrect comparison by changing the value type or the comparison
#5	MilkyWayExToken	Tautology or contradiction (tautology)	419, 425	Fix the incorrect comparison by changing the value type or the comparison

## Informational issues

Issue	File	Type	Line	Description
#1	MilkyWayExToken	State variables that could be declared constant (constable-states)	47, 38, 36, 29, 28, 40, 39	Add the `constant` attributes to state variables that never change
#2	MilkyWayExRewardsTracker	State variables that could be declared constant (constable-states)	37	Add the `constant` attributes to state variables that never change
#3	RewardsContract	State variables that could be declared constant (constable-states)	18	Add the `constant` attributes to state variables that never change

## Audit Comments

### 22. October 2021:

#### v1.0

- Deployer can lock user funds
  - If address is added to preventer mapping in RewardsContract.sol
- Deployer can lock dividends if address is added to excludedFromDividends mapping with excludeFromDividends function
- The return value of a low-level call is not checked and may lead to wrong results

#### v1.1

- Fixed medium issue

## SWC Attacks

ID	Title	Relationships	Status
<a href="#">SW C-13 6</a>	Unencrypted Private Data On-Chain	<a href="#">CWE-767: Access to Critical Private Variable via Public Method</a>	PASSED
<a href="#">SW C-13 5</a>	Code With No Effects	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-13 4</a>	Message call with hardcoded gas amount	<a href="#">CWE-655: Improper Initialization</a>	PASSED
<a href="#">SW C-13 3</a>	Hash Collisions With Multiple Variable Length Arguments	<a href="#">CWE-294: Authentication Bypass by Capture-replay</a>	PASSED
<a href="#">SW C-13 2</a>	Unexpected Ether balance	<a href="#">CWE-667: Improper Locking</a>	PASSED
<a href="#">SW C-13 1</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-13 0</a>	Right-To-Left-Override control character (U+202E)	<a href="#">CWE-451: User Interface (UI) Misrepresentation of Critical Information</a>	PASSED
<a href="#">SW C-12 9</a>	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	PASSED
<a href="#">SW C-12 8</a>	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	PASSED

<a href="#">SW C-12 7</a>	Arbitrary Jump with Function Type Variable	<a href="#">CWE-695: Use of Low-Level Functionality</a>	<b>PASSED</b>
<a href="#">SW C-12 5</a>	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	<b>PASSED</b>
<a href="#">SW C-12 4</a>	Write to Arbitrary Storage Location	<a href="#">CWE-123: Write-what-where Condition</a>	<b>PASSED</b>
<a href="#">SW C-12 3</a>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	<b>PASSED</b>
<a href="#">SW C-12 2</a>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	<b>PASSED</b>
<a href="#">SW C-12 1</a>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>
<a href="#">SW C-12 0</a>	Weak Sources of Randomness from Chain Attributes	<a href="#">CWE-330: Use of Insufficiently Random Values</a>	<b>PASSED</b>
<a href="#">SW C-11 9</a>	Shadowing State Variables	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-11 8</a>	Incorrect Constructor Name	<a href="#">CWE-665: Improper Initialization</a>	<b>PASSED</b>
<a href="#">SW C-11 7</a>	Signature Malleability	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>

<a href="#">SW C-11 6</a>	Timestamp Dependence	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	PASSED
<a href="#">SW C-11 5</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	PASSED
<a href="#">SW C-11 4</a>	Transaction Order Dependence	<a href="#">CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')</a>	PASSED
<a href="#">SW C-11 3</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	PASSED
<a href="#">SW C-11 2</a>	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	PASSED
<a href="#">SW C-111</a>	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	PASSED
<a href="#">SW C-11 0</a>	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	PASSED
<a href="#">SW C-10 9</a>	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	PASSED
<a href="#">SW C-10 8</a>	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	PASSED
<a href="#">SW C-10 7</a>	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	PASSED
<a href="#">SW C-10 6</a>	Unprotected SELFDESTRUCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	PASSED

<a href="#">SW C-10 5</a>	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>
<a href="#">SW C-10 4</a>	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	<b>PASSED</b>
<a href="#">SW C-10 3</a>	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	<b>NOT PASSED</b>
<a href="#">SW C-10 2</a>	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	<b>PASSED</b>
<a href="#">SW C-10 1</a>	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	<b>PASSED</b>
<a href="#">SW C-10 0</a>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>

The logo features the words "Solid Proofed" in a white, elegant script font. The word "Solid" is positioned above "Proofed". Behind the text is a faint, stylized shield emblem with a grid-like pattern, rendered in a darker shade of blue. The entire composition is set against a solid blue background.

Solid  
Proofed

**Blockchain Security | Smart Contract Audits | KYC**

A small horizontal representation of the German flag, consisting of three equal-width horizontal stripes of black, red, and gold.

MADE IN GERMANY