



# SOLIDProof

*Bring trust into your projects*

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

MADE IN GERMANY

# Virgo

# Audit

**Security Assessment**  
**22. March, 2023**

**For**



**SolidProof\_io**



**@solidproof\_io**

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
Inheritance Graph	13
CallGraph	14
Scope of Work/Verify Claims	15
Modifiers and public functions	25
Source Units in Scope	29
Critical issues	30
High issues	30
Medium issues	30
Low issues	30
Informational issues	30
Audit Comments	30
SWC Attacks	31

# 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 analyse.

Version	Date	Description
1.0	10 March 2023 to 15. March 2023	<ul style="list-style-type: none"><li>• Layout project</li><li>• Automated- /Manual-Security Testing</li><li>• Summary</li></ul>
1.1	22. March 2023	<ul style="list-style-type: none"><li>• Reaudit</li></ul>

## **Network**

Ethereum

## **Website**

<https://messier.app/>

## **Telegram**

<https://t.me/MessierM87Community>

## **Twitter**

<https://twitter.com/MessierM87>

## **Medium**

<https://github.com/MessierM87>



## Description

Our main goal is to create decentralized applications that will provide both consumers and businesses with tools designed to make cryptocurrency transactions more confidential, secure, and viable than conventional currencies.

The dapps that Messier creates, are aimed at the general public, but their specific use cases may vary. Generally, the main entities that will want to use our dapps are individuals or organizations looking to utilize decentralized applications for a variety of purposes within the field of financial transactions.

## Project Engagement

During the Date of 09 March 2023, **Messier Team** engaged Solidproof.io to audit smart contracts that they created for the **Virgo DAPP**. 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 white paper.

## Logo



VIRGO

## Contract Link

### v1.0

- [https://github.com/MessierM87/VIRGO\\_SolidityContracts](https://github.com/MessierM87/VIRGO_SolidityContracts)
- Commit: d98832a

### v1.1

- [https://github.com/MessierM87/VIRGO\\_SolidityContracts](https://github.com/MessierM87/VIRGO_SolidityContracts)
- Commit: fdee5d3

# 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:

Dependency / Import Path	Count
@openzeppelin/contracts-upgradeable/access/IAccessControlEnumerableUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	1
@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol	2
@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol	1
@openzeppelin/contracts-upgradeable/utils/CountersUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/utils/structs/EnumerableSetUpgradeable.sol	1
@openzeppelin/contracts/access/AccessControlEnumerable.sol	1
@openzeppelin/contracts/access/Ownable.sol	4
@openzeppelin/contracts/token/ERC20/ERC20.sol	5
@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol	1
@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol	1
@openzeppelin/contracts/utils/Context.sol	1
@openzeppelin/contracts/utils/math/SafeMath.sol	1
@uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol	1
@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol	1
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol	1
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol	2



## 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

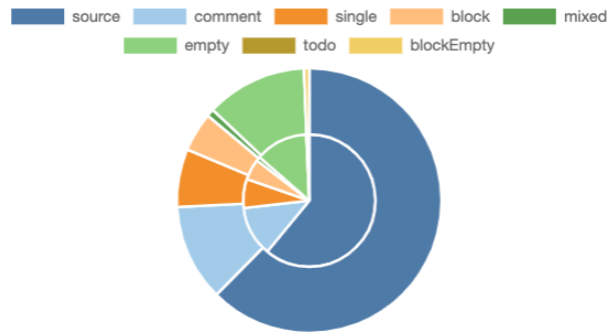
File Name	SHA-1 Hash
contracts/MarketPlace.sol	ca9645b3c61964ba9fdc8effe33d949c12366aff
contracts/libs/M87Bank.sol	6725300982978df01e0377b5db62d78c46c432cf
contracts/libs/xToken.sol	ebd81a8a8c56697b06943faf1d1d9a65cc95202b
contracts/libs/IUniSwap.sol	c48281fe667a22a0dad3699022f29c65390dd69
contracts/libs/MESSIER.sol	b3bf8f3a3e0ddc0dceae1ddb54441f4720345611
contracts/libs/MOTTToken.sol	92f4bb1e4e0f72bac2791f8839dec2d1b55f7d6b
contracts/libs/IMessierNFT.sol	5032fef30a77caa6f2f8af75a0d81b075466a452
contracts/libs/simulatorM87.sol	3d448ced3ae3cb98595904b68700850dd1ddf7c7
contracts/libs/MTTToken.sol	11c732dd5fc459d32365685260edfefe0d435365
contracts/libs/MessierNFT.sol	a80083baf52800020307572ff57108f7f773e9a0
contracts/libs/Pausable.sol	b94e4418d6caa6b2cc601ea21faee00134deed01

contracts/libs/IM87.sol	9247fbca437a2aa4826abc884a56153f6551f3fd
contracts/libs/ ISupernova.sol	3e1582c7baee6e4210d00295c4c53a50e9f9c1f8
contracts/Supernova.sol	8b3628dec116c114eea5b6db337e214209b58e21
contracts/Dao.sol	f2dd90cc440304545314c334f67898474f9097d6

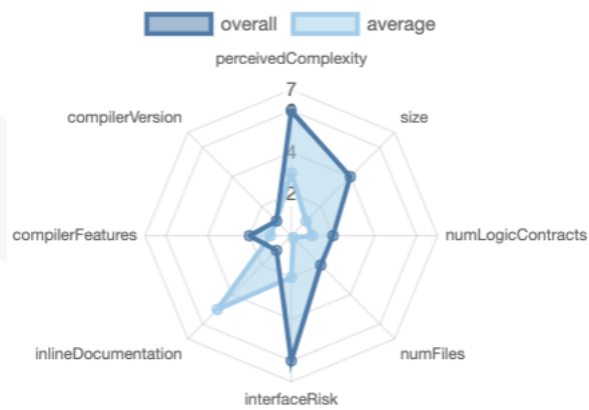


# Metrics

## Source Lines v1.0



## Risk Level v1.0



# Capabilities

## Components

 Contracts	 Libraries	 Interfaces	 Abstract
13	0	4	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.












 Public	 Payable
161	15

External	Internal	Private	Pure	View
81	177	15	0	61

### StateVariables

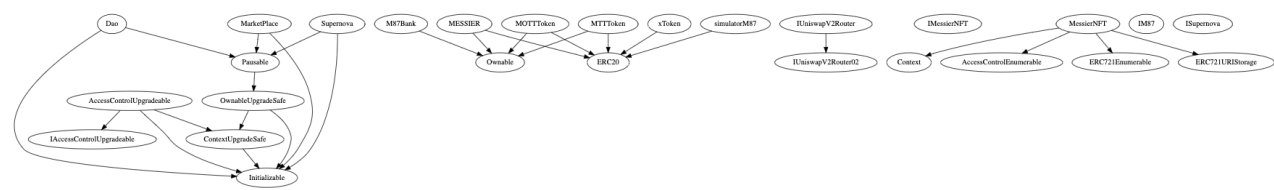
Total	 Public
137	39

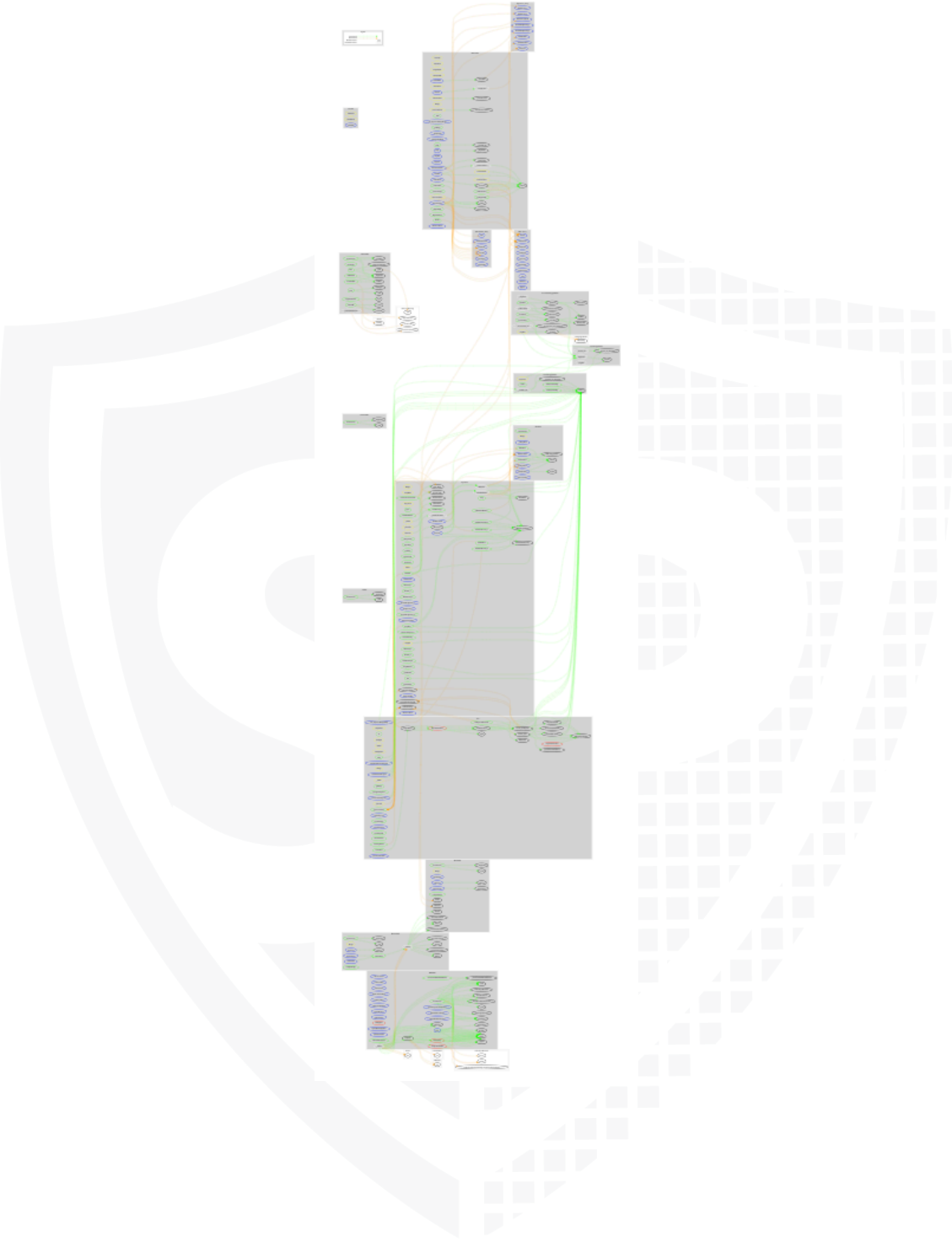
### Capabilities

Solidity Versions observed	 Experimental Features	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts	
 0.8.9		yes			
 Transfers ETH	 Low-Level Calls	 DelegateCall	 Uses Hash Functions	 ECRrecover	 New/Create/Create2
yes			yes		yes → NewContract:M87Bank

# Inheritance Graph

## 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	
Are any contracts upgradeable?	Yes

Comments:

### v1.0

- Owner can deploy a new version of the **Supernova and Pausable** contracts which can change any limit and give owner new privileges
  - Be aware of this and do your own research for the contract which is the contract pointing to



## 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	✓	✓	✓

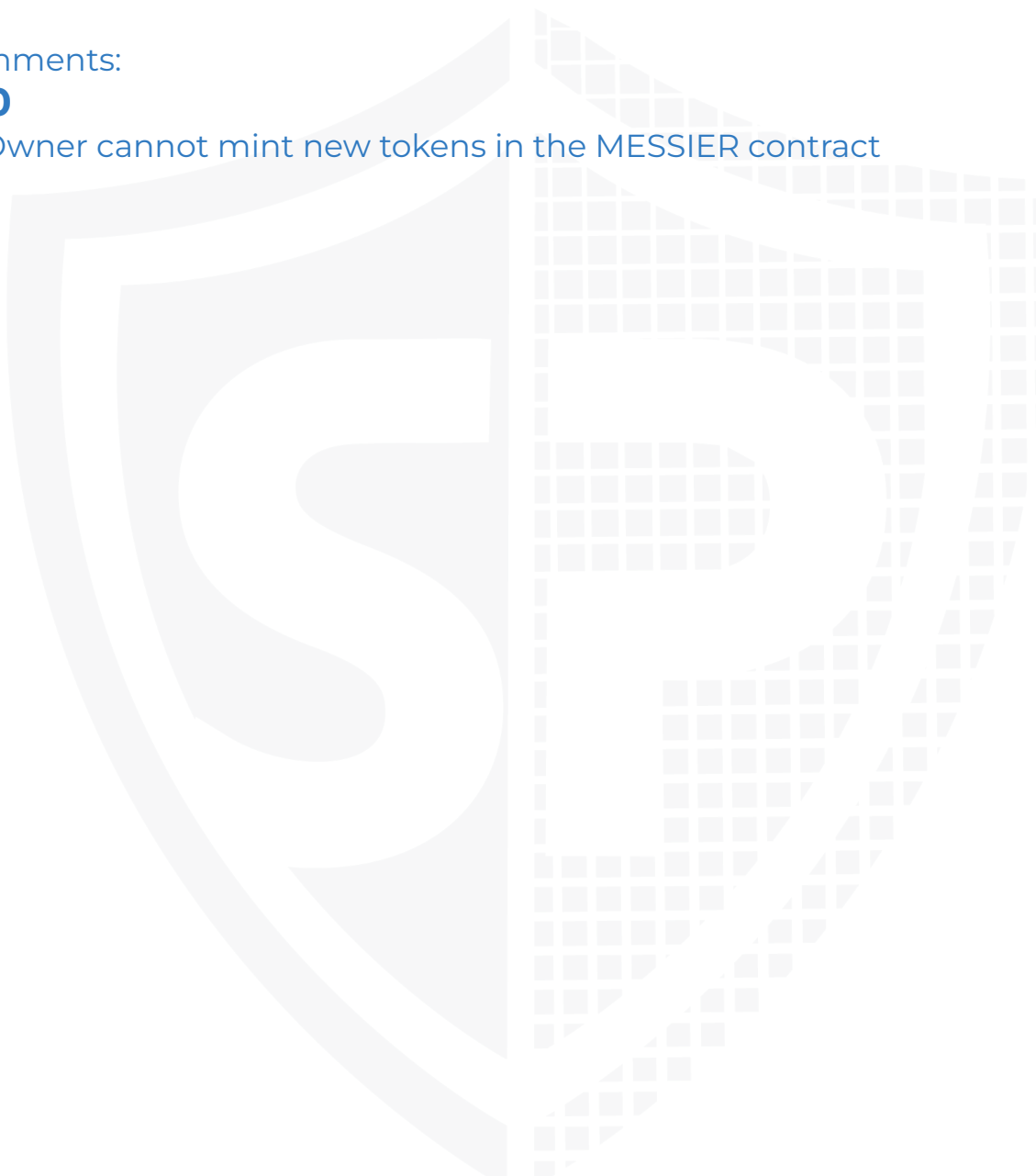
## Deployer cannot mint any new tokens

Name	Exist	Tested	Status
Deployer cannot mint	✓	✓	✓
Max / Total Supply	1.000.000.000.000		

Comments:

### v1.0

- Owner cannot mint new tokens in the MESSIER contract



## Deployer cannot burn or lock user funds

Name	Exist	Tested	Status
Deployer cannot lock	—	—	—
Deployer cannot burn	—	—	—



## 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

- Fees cannot be set without any limitations in the 'MESSIER.sol' contract

## Deployer can blacklist/antisnipe addresses

Name	Exist	Tested	Status
Deployer can blacklist/antisnipe addresses	✓	✓	✗

Comments:

**v1.0**

- Owner is able to blacklist addresses in the 'MESSEIR.sol' contract



## 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

## M87Bank.sol

- ◆ \_ChangeHash
- ◆ WithdrawEth 💰
- ◆ WithdrawToken 💰

## MESSIER.sol


- ◆ enableTrading
- Ⓜ onlyOwner
- ◆ removeLimits
- Ⓜ onlyOwner
- ◆ disableTransferDelay
- Ⓜ onlyOwner
- ◆ setEarlySellTax
- Ⓜ onlyOwner
- ◆ updateSwapTokensAtAmount
- Ⓜ onlyOwner
- ◆ updateMaxTxnAmount
- Ⓜ onlyOwner
- ◆ updateMaxWalletAmount
- Ⓜ onlyOwner
- ◆ excludeFromMaxTransaction
- Ⓜ onlyOwner
- ◆ updateSwapEnabled
- Ⓜ onlyOwner
- ◆ updateBuyFees
- Ⓜ onlyOwner
- ◆ updateSellFees
- Ⓜ onlyOwner
- ◆ excludeFromFees
- Ⓜ onlyOwner
- ◆ setAutomatedMarketMakerPair
- Ⓜ onlyOwner
- ◆ updateMarketingWallet
- Ⓜ onlyOwner
- ◆ updateDevWallet
- Ⓜ onlyOwner
- ◆ Send
- Ⓜ onlyOwner

## DAO.sol

- ◆ setup
- Ⓜ initializer
- ◆ \_Set\_SupernovaBridgeBridge
- Ⓜ onlyOwner
- ◆ createCashOutProposal
- Ⓜ \_IsPowehi
- ◆ createDoubleClassicProposal
- Ⓜ \_IsPowehi
- ◆ createStructureProposal
- Ⓜ \_IsPowehi
- ◆ signOnProposal
- Ⓜ \_IsPowehi
- Ⓜ notSigner
- Ⓜ atState
- ◆ VoteOnProposal
- Ⓜ \_IsHalo
- Ⓜ stateQuorum
- Ⓜ atState
- ◆ CloseProposal
- Ⓜ atState
- Ⓜ OnlyOracle
- ◆ OracleNext
- Ⓜ OnlyOracle
- ◆ FullCycleToExecution

## MarketPlace.sol

- ◆ setup
- Ⓜ initializer
- ◆ \_Set\_SupernovaBridgeBridge
- Ⓜ onlyOwner
- ◆ \_ChangeHash
- Ⓜ onlyOwner
- ◆ \_ChangeStateManul
- Ⓜ onlyOwner
- ◆ makeBidInit
- Ⓜ correctId
- Ⓜ auctionInit
- Ⓜ minimumBid
- Ⓜ balanceOfToken
- ◆ RefundsEth
- ◆ DropEth
- ◆ Drop
- Ⓜ correctId
- Ⓜ auctionInit
- Ⓜ dropOfToken
- ◆ DetermineWinInit
- ◆ Refunds
- ◆ Resell
- ◆ makeBid 💰
- ◆ AddFundEth 💰
- ◆ AddFundToken 💰
- Ⓜ balanceOfToken
- ◆ DetermineWin
- ◆ <Constructor> 💰



- ◆ setup
- Ⓜ initializer
- ◆ changeOracle
- Ⓜ onlyOwner
- ◆ InjectHalo
- Ⓜ OnlyOracle
- ◆ InjectPowehi
- Ⓜ OnlyOracle
- ◆ returnContractCurrentId
- ◆ \_WithdrawToken 💰
- ◆ WithdrawEth 💰
- Ⓜ Bridge
- ◆ PutInReward\_1
- Ⓜ Bridge
- ◆ PutInReward\_2
- Ⓜ Bridge
- ◆ PutAndDropReward\_1
- Ⓜ Bridge
- ◆ DropReward\_1
- Ⓜ Bridge
- ◆ PutAndDropReward\_2
- Ⓜ Bridge
- ◆ PutInTreasuryETH 💰
- Ⓜ Bridge
- ◆ PutInTreasuryet 💰
- ◆ PutInTreasuryToken
- ◆ PutOutTreasury
- Ⓜ Bridge
- ◆ PutOutTokenTreasuryB
- Ⓜ Bridge
- ◆ StakM87
- ◆ PutDarkList
- ◆ DropDarkList
- ◆ DropM87
- ◆ WithdrawReward\_1
- ◆ WithdrawReward\_2
- ◆ Received
- Ⓜ Bridge

## Ownership/Authority Privileges

S.No	File	Privileges
#1	MESSIER.sol	<ul style="list-style-type: none"> <li>• Enable trading and transfer delay but cannot disable it</li> <li>• Remove limits and cannot add them again</li> <li>• Enable/Disable early sell tax</li> <li>• Update the amount to swap tokens, max transaction amount, and max wallet amount within a safe range</li> <li>• Set fees but it cannot be more than 25%</li> <li>• Include/Exclude wallets from max transaction amount, and fees</li> <li>• Update marketing, and dev wallet address</li> <li>• Set AMM pair address</li> </ul>
#2	MOTTToken.sol	<ul style="list-style-type: none"> <li>• Users can only transfer tokens if they pass the right hash value in the transfer function.</li> <li>• The Hash value will be constant after deployment.</li> </ul>
#3	M87Bank.sol	<ul style="list-style-type: none"> <li>• The owner is able to set/change the hash in the contract at any time</li> <li>• The users who have access to the hash can withdraw any type of tokens, if available in the contract balance.</li> </ul>
#4	DAO.sol (isPowehi addresses, OnlyOracle and owner)	<ul style="list-style-type: none"> <li>• Set the address of Supernova bridge</li> <li>• Create cash out proposal</li> <li>• Create Structure proposal</li> <li>• Only _IsPowehi addresses can sign on proposal.</li> <li>• Only _isHalo addresses can vote on proposals</li> <li>• Only Oracle address can close and checkOut a proposal</li> <li>• Only Oracle address can execute cycles</li> </ul>
#5	Marketplace.sol	<ul style="list-style-type: none"> <li>• Set the address of Supernova bridge</li> <li>• Change HASH value, and state</li> </ul>

S.No	File	Privileges
#6	Supernova.sol (Owner, onlyOracle)	<ul style="list-style-type: none"> <li>• Change oracle address</li> <li>• Oracle address can Inject halo addresses (and only these addresses can vote in the Dao) , and Power addresses</li> <li>• Users with access to the HASH can: <ul style="list-style-type: none"> <li>- Withdraw ETH, and token</li> <li>- Put in rewards</li> <li>- Drop rewards</li> <li>- Put out treasury</li> </ul> </li> </ul>

- There are several authorities which are authorized to call some functions, that means, if the owner is renounced, another address is still authorized to call functions
  - Be aware of this

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

# Source Units in Scope

## v1.0

File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score
contracts/MarketPlace.sol	1	————	579	533	427	40	227
contracts/libs/M87Bank.sol	1	————	57	57	51	1	48
contracts/libs/xToken.sol	1	————	14	14	8	1	6
contracts/libs/IUniSwap.sol	————	1	9	9	6	1	3
contracts/libs/MESSIER.sol	1	————	498	491	288	125	249
contracts/libs/MOTTToken.sol	1	————	90	67	50	3	41
contracts/libs/IMessierNFT.sol	————	1	18	6	3	1	13
contracts/libs/simulatorM87.sol	1	————	14	14	8	1	6
contracts/libs/MTTToken.sol	1	————	87	64	43	16	36
contracts/libs/MessierNFT.sol	1	————	123	96	68	14	59
contracts/libs/Pausable.sol	4	————	295	295	153	102	106
contracts/libs/IM87.sol	————	1	31	10	5	1	21
contracts/libs/ISupernova.sol	————	1	15	6	3	1	23
contracts/Supernova.sol	1	————	731	718	547	80	356
contracts/Dao.sol	1	————	942	859	679	101	368
<b>Totals</b>	<b>14</b>	<b>4</b>	<b>3503</b>	<b>3239</b>	<b>2339</b>	<b>488</b>	<b>1562</b>

## 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	All	A floating pragma is set	—	The current pragma Solidity directive is „^0.8.9“.

## Informational issues

Issue	File	Type	Line	Description
#1	All	NatSpec documentation missing	—	If you started to comment your code, also comment all other functions, variables etc.

## 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.

### 22. March 2023:

- There is still an owner (Owner still has not renounced ownership)
- Owner can deploy a new version of the contract which can change any limit and give owner new privileges
- Read whole report and modifiers section for more information

## SWC Attacks

ID	Title	Relationships	Status
<a href="#">SW C-1 36</a>	Unencrypted Private Data On-Chain	<a href="#">CWE-767: Access to Critical Private Variable via Public Method</a>	PASSED
<a href="#">SW C-1 35</a>	Code With No Effects	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-1 34</a>	Message call with hardcoded gas amount	<a href="#">CWE-655: Improper Initialization</a>	PASSED
<a href="#">SW C-1 33</a>	Hash Collisions With Multiple Variable Length Arguments	<a href="#">CWE-294: Authentication Bypass by Capture-replay</a>	PASSED
<a href="#">SW C-1 32</a>	Unexpected Ether balance	<a href="#">CWE-667: Improper Locking</a>	PASSED
<a href="#">SW C-1 31</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-1 30</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-1 29</a>	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	PASSED
<a href="#">SW C-1 28</a>	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	PASSED

<a href="#">SW C-1 27</a>	Arbitrary Jump with Function Type Variable	<a href="#">CWE-695: Use of Low-Level Functionality</a>	<b>PASSED</b>
<a href="#">SW C-1 25</a>	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	<b>PASSED</b>
<a href="#">SW C-1 24</a>	Write to Arbitrary Storage Location	<a href="#">CWE-123: Write-what-where Condition</a>	<b>PASSED</b>
<a href="#">SW C-1 23</a>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	<b>PASSED</b>
<a href="#">SW C-1 22</a>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	<b>PASSED</b>
<a href="#">SW C-1 21</a>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>
<a href="#">SW C-1 20</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>	<b>PASSED</b>
<a href="#">SW C-11 5</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<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>	<b>PASSED</b>
<a href="#">SW C-11 3</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	<b>PASSED</b>
<a href="#">SW C-11 2</a>	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-11 1</a>	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 0</a>	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	<b>PASSED</b>
<a href="#">SW C-1 09</a>	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	<b>PASSED</b>
<a href="#">SW C-1 08</a>	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-1 07</a>	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	<b>PASSED</b>
<a href="#">SW C-1 06</a>	Unprotected SELFDESTRUCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>

<a href="#">SW C-1 05</a>	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>
<a href="#">SW C-1 04</a>	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	<b>PASSED</b>
<a href="#">SW C-1 03</a>	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	<b>NOT PASSED</b>
<a href="#">SW C-1 02</a>	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	<b>PASSED</b>
<a href="#">SW C-1 01</a>	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	<b>PASSED</b>
<a href="#">SW C-1 00</a>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>

*Solid  
Proofed*

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

  
MADE IN GERMANY