



SOLIDProof
Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

v1.0: 27. January, 2022

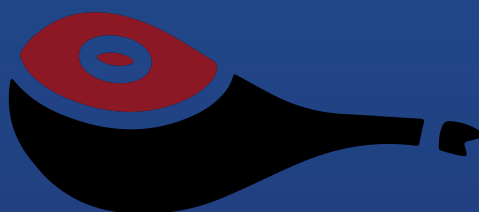
Jamonswap

Audit

Security Assessment

03. February, 2022

For



JAMONSWAP

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	23
Source Units in Scope	28
Critical issues	30
High issues	30
Medium issues	30
Low issues	30
Informational issues	30
Audit Comments	32
SWC Attacks	33

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	26. January 2022	<ul style="list-style-type: none">• Layout project• Automated- /Manual-Security Testing• Summary
	27. January 2022	Finished
1.1	03. February 2022	Reaudit

Network

Polygon

Website

<https://jamonswap.finance/>

Telegram

<https://t.me/+4TWEeg5uQX02YTlk>

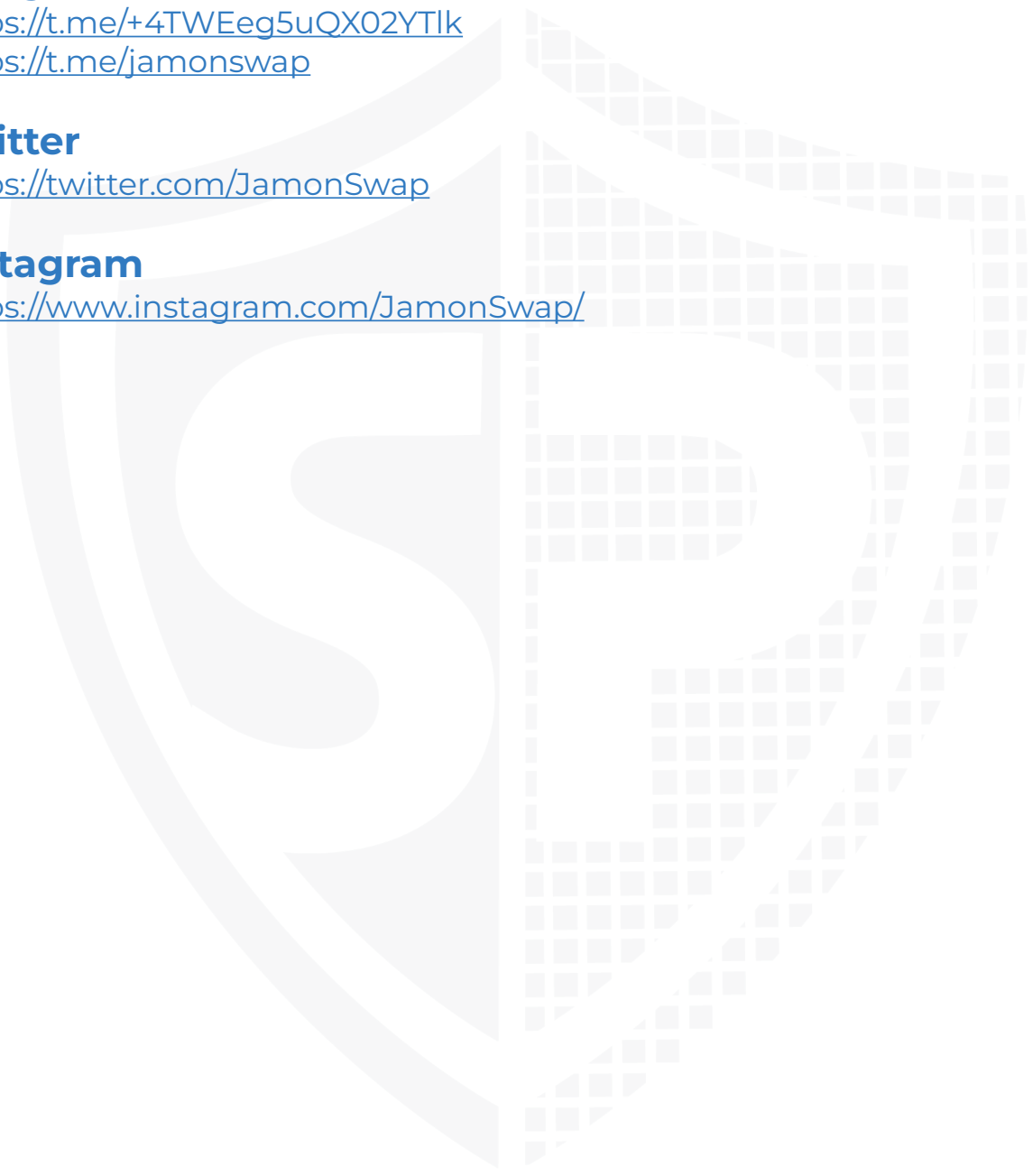
<https://t.me/jamonswap>

Twitter

<https://twitter.com/JamonSwap>

Instagram

<https://www.instagram.com/JamonSwap/>



Description

TBA

Project Engagement

During the 24th of January 2022, **Jamonswap 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



JAMONSWAP

Contract Link

v1.0

- Github
 - <https://github.com/jamonswap/contracts>
 - Commit: c9a769356633bfeb8b921cae33b3b19059af74a8

v1.1

- Github
 - <https://github.com/jamonswap/contracts>
 - Commit
 - 158636883b0f2ef9da9c44ffbac84a1050af1b2f
 - 6ae1e22b9c576b2b92c7b456b62e38aa004a83fe

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 as possible.
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
@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol	2
@openzeppelin/contracts/access/AccessControl.sol	2
@openzeppelin/contracts/access/Ownable.sol	8
@openzeppelin/contracts/governance/Governor.sol	1
@openzeppelin/contracts/governance/extensions/GovernorCountingSimple.sol	1
@openzeppelin/contracts/governance/extensions/GovernorSettings.sol	1
@openzeppelin/contracts/governance/extensions/GovernorVotes.sol	1
@openzeppelin/contracts/governance/extensions/GovernorVotesQuorumFraction.sol	1
@openzeppelin/contracts/security/Pausable.sol	7
@openzeppelin/contracts/security/ReentrancyGuard.sol	7
@openzeppelin/contracts/token/ERC20/ERC20.sol	2
@openzeppelin/contracts/token/ERC20/IERC20.sol	9
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	2
@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol	1
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	1
@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol	2
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	5
@openzeppelin/contracts/utils/Counters.sol	3
@openzeppelin/contracts/utils/math/Math.sol	1
@openzeppelin/contracts/utils/math/SafeMath.sol	7
@openzeppelin/contracts/utils/structs/EnumerableSet.sol	4

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

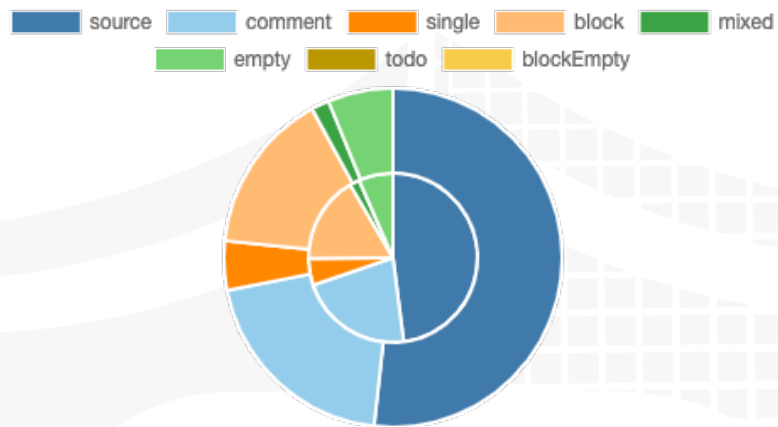
File Name	SHA-1 Hash
contracts/interfaces/IJamonSharePresale.sol	0f0fe57831b17ffe9736313883f0379c1666596c
contracts/interfaces/IConversor.sol	c4d2abb95bd34c3d17e3c403d88d11bf6e756136
contracts/interfaces/IJamonVesting.sol	c9956a482deb7e04240129136a9a2928a33b984f
contracts/interfaces/IJamonShareVesting.sol	6b49e665293c6ab006c2ec4d1fb2c68618e3aa46
contracts/interfaces/IERC20MintBurn.sol	304ea9d0e6ffaebef6c6cf9a004ad55fd31235d6
contracts/interfaces/IJamonRouter.sol	a38638b24ff3b67fdf8bf37368b796e5900e1dac
contracts/interfaces/IJamonPair.sol	0efc85f7bf717969dd5e8a1e7faf22b5b613a1f8
contracts/tokens/JamonShare.sol	3b7c957fd74c78c7767a9ede34729a3bf09c82e5
contracts/tokens/JamonV2.sol	d1e532fb8bc8fc9d6c4e97c7abd964161fc6905
contracts/JamonVesting.sol	5c74ea00b3fc96e6ec41d3a55f3bd9fb2c04d897
contracts/JamonShareVault.sol	9f253d6b5ecad903bc90f2ff12f6cacfd91356e5
contracts/Conversor.sol	a5d075b4f0897e07ede25135d2b11f338e359e9e
contracts/JamonSharePresale.sol	a4858bde72be967f138a2dc40fa55f730164d965
contracts/Bonus.sol	d737bc58ee04f210475c4bb1640297ed8bcbbd05
contracts/JamonShareVesting.sol	8fff8ac326aa9d5034054dbbbae5b559c40e4dff
contracts/JamonGovernor.sol	7e36ffc32cce51f19e1b059e3748407d28d32899
contracts/JamonVault.sol	9b6a496d16d29d780fbb1e82215e16db6e54deee

V1.1

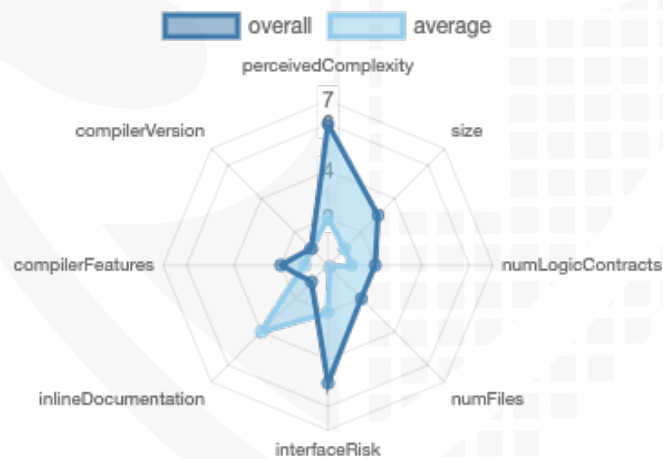
File Name	SHA-1 Hash
contracts/interfaces/IJamonSharePresale.sol	5d4a5717f46e3bec98b1aa16d5941f54746b5706
contracts/interfaces/IConversor.sol	6fee460616b439fb9d7e9844956c8dc3152e2078
contracts/interfaces/IJamonVesting.sol	b5973db3a871c85916b50e55afa9ff7097464d99
contracts/interfaces/IJamonShareVesting.sol	006b77d762cff21271a9842ff9f77d81c34035d
contracts/interfaces/IERC20MintBurn.sol	3cb2cf54d48448e8dbe02e7aaf1580cf1ef16db8
contracts/interfaces/IJamonRouter.sol	9c1483ae3e2c661f2209afb45681e7ce30538fec
contracts/interfaces/IJamonPair.sol	6ac768acd89fdc397061ac80ea8eb062c1d659fc
contracts/tokens/JamonShare.sol	50c205d0426de2658f7b31626d005faa0841ffce
contracts/tokens/JamonV2.sol	476710f3e95c56e7e50dc57a7f9818f4ef2b6f9d
contracts/JamonVesting.sol	638c51e3ee380f309adf740ad923baa1cd528223
contracts/JamonShareVault.sol	d2bf400aacbde0912dc287549f5bc82d3a4a6da6
contracts/Conversor.sol	4e9059850f6b338164d1c0a7e46764d22260a12
contracts/JamonSharePresale.sol	0cd4943da095228703ff9dedea646a8d8dc8b512
contracts/Bonus.sol	80315cd5446fcae1ccc9c33aa1341b6560d04386
contracts/JamonShareVesting.sol	4b068dfaf4747b7070c7bb2bc16c8c9cb345a6fd
contracts/JamonGovernor.sol	34885e063a83ac69a9078ac27c10c32b44846ccf
contracts/JamonVault.sol	c5480c0fb2f41c9d979e15113d15dee1b246de7f

Metrics

Source Lines v1.0



Risk Level v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	10	0	8	0

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	122	0
1.1	133	0

Version	External	Internal	Private	Pure	View
1.0	87	142	0	3	53
1.1	95	146	0	3	58

State Variables

Version	Total	Public
1.0	73	19
1.1	79	25

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	^0.8.11				

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	EC Recover	New/Create/Create2
1.0	yes			yes		

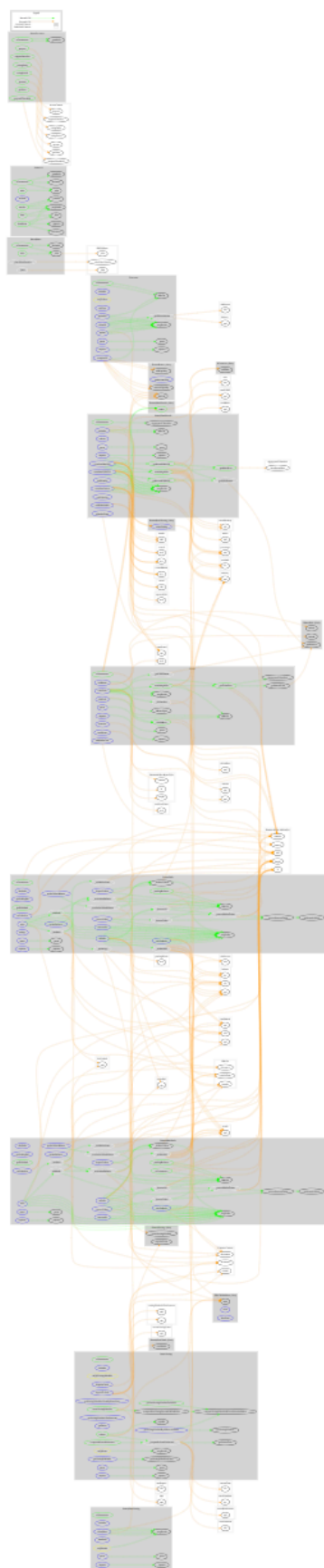
Inheritance Graph

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

Correct implementation of Token standard

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 v1.0

▼ JAMONV2	▼ JAMONSHARE	▼ BONUS
approve	approve	addBonus
burn	burn	addFeed
burnFrom	burnFrom	addStableCoin
decreaseAllowance	decreaseAllowance	contribute
grantRole	delegate	pause
increaseAllowance	delegateBySig	renounceOwnership
mint	increaseAllowance	transferOwnership
permit	mint	unpause
renounceRole	permit	
revokeRole	renounceOwnership	
setVault	transfer	
transfer	transferFrom	
transferFrom	transferOwnership	

▼ CONVERSOR	▼ JAMONGOVERNOR	▼ JAMONSHAREVAULT
claimLP	castVote	depositTokens
completeLP	castVoteBySig	harvestAll
initialize	castVoteWithReason	harvestToken
pause	execute	pause
renounceOwnership	grantRole	renounceOwnership
transferOwnership	propose	safeUnStake
unpause	renounceRole	setTokenList
update	revokeRole	stake
updateLP	setProposalThreshold	transferOwnership
	setVotingDelay	unpause
	setVotingPeriod	unStake
	updateQuorumNumerator	updateBalances
		updateTokenBalance

▼ JAMONSHAREVESTING

- claimShare
- createVesting
- initialize
- pause
- renounceOwnership
- transferOwnership
- unpause

▼ JAMONVESTING

- createVestingSchedule
- depositToVault
- initialize
- pause
- release
- renounceOwnership
- transferOwnership
- unpause

▼ JAMONVAULT

- depositTokens
- harvestAll
- harvestToken
- pause
- renounceOwnership
- safeUnStake
- setApy
- setTokenList
- stake
- transferOwnership
- unpause
- unStake
- updateBalances
- updateTokenBalance

▼ JAMONSHAREPRESALE

- contributeJamon
- contributeMaticLP
- contributeUSDCLP
- editWhitelist
- editWhitelistLP
- initialize
- pause
- renounceOwnership
- setWhitelist
- transferOwnership
- unpause

Cannot mint any new tokens

Name	Exist	Tested	Status
Cannot mint	✓	✓	✗

Comments:

v1.0

- Conversor
 - update function will mint new tokens (convert the old token for the new one)
 - onlyOwner can call completeLP function to mint new token (change old liquidity to new liquidity)
- JamonShareVesting
 - claimShare will mint new tokens
 - createVesting can only be called from presale and mint new tokens
- JamonVault
 - updateBalances will mint new tokens
- JamonVesting
 - Release will mint new tokens (only if vesting schedules)
 - depositToVault will mint new tokens to shareVault address
- Every contract which is using IERC20MintBurn interface can mint/burn

Cannot burn or lock user funds

Name	Exist	Tested	Status
Cannot lock	✓	✓	✗
Cannot burn	✓	✓	✗

Comments:

v1.0

- JamonSharePresale
 - contributeJamon function will burn from caller
 - onlyOwner can lock contributeMaticLP function if round equal to 1 and user is not in WhitelistLP
 - Deployer can lock contributeJamon function by setting a Max4Wallet amount to an specific address
- Every contract which is using IERC20MintBurn interface can mint/burn

Deployer cannot pause the contract

Name	Exist	Tested	Status
Deployer cannot pause	✓	✓	✗

Comments:

v1.0

- Deployer can lock following functions with pausing the contract
 - Bonus
 - contribute
 - Conversor
 - updateLP
 - Update
 - claimLP
 - JamonSharePresale
 - contributeMaticLP
 - contributeUSDCLP
 - contributeJamon
 - JamonShareVault
 - Stake
 - harvestToken
 - harvestAll
 - unStake
 - updateBalance
 - JamonShareVesting
 - createVesting
 - JamonVault
 - Stake
 - harvestToken
 - harvestAll
 - unStake
 - updateBalance
 - JamonVesting
 - createVestingSchedule
 - Release
 - depositToVault

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

JamonShare

▼ 🔹 mint

Ⓜ onlyOwner

🔹 transfer

🔹 approve

🔹 transferFrom

🔹 increaseAllowance

🔹 decreaseAllowance

🔹 burn

🔹 burnFrom

▼ 🔹 renounceOwnership

Ⓜ onlyOwner

▼ 🔹 transferOwnership

Ⓜ onlyOwner

🔹 permit

🔹 delegate

🔹 delegateBySig

JamonV2

▼ 🔹 mint

Ⓜ onlyRole

▼ 🔹 setVault







Ⓜ onlyRole

🔹 transfer








🔹 burn

🔹 burnFrom

Bonus

- ✓  **contribute**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓  **addBonus**
 - Ⓜ onlyOwner
- ✓  **addFeed**
 - Ⓜ onlyOwner
- ✓  **addStableCoin**
 - Ⓜ onlyOwner
- ✓  **pause**
 - Ⓜ onlyOwner
- ✓  **unpause**
 - Ⓜ onlyOwner

Conversor

- ✓  **initialize**
 - Ⓜ onlyOwner
- ✓  **updateLP**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
 - Ⓜ onlyTokens
- ✓  **update**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓  **claimLP**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓  **completeLP**
 - Ⓜ onlyOwner
- ✓  **pause**
 - Ⓜ onlyOwner
- ✓  **unpause**
 - Ⓜ onlyOwner

JamonGovernor

- grantRole
 - onlyRole
- revokeRole
 - onlyRole
- renounceRole

- setVotingDelay
 - onlyGovernance
- setVotingPeriod
 - onlyGovernance
- setProposalThreshold
 - onlyGovernance

- propose
 - onlyRole

- updateQuorumNumerator
 - onlyGovernance

- propose
- execute 💰
- castVote
- castVoteWithReason
- castVoteBySig

JamonSharePresale

- initialize
 - onlyOwner
- contributeMaticLP
 - whenNotPaused
 - nonReentrant
- contributeUSDCLP
 - whenNotPaused
 - nonReentrant
- contributeJamon
 - whenNotPaused
 - nonReentrant
- setWhitelist
 - onlyOwner
- editWhitelistLP
 - onlyOwner
- editWhitelist
 - onlyOwner
- pause
 - onlyOwner
- unpause
 - onlyOwner

JamonShareVault

- depositTokens
 - nonReentrant
- stake
 - whenNotPaused
 - nonReentrant
- harvestToken
 - whenNotPaused
 - nonReentrant
- harvestAll
 - whenNotPaused
 - nonReentrant
- unStake
 - whenNotPaused
 - nonReentrant
- safeUnStake
 - whenPaused
 - nonReentrant
- updateBalances
 - whenNotPaused
 - nonReentrant
- updateTokenBalance
 - onlyOwner
- setTokenList
 - onlyOwner
- pause
 - onlyOwner
- unpause
 - onlyOwner

JamonShareVesting

- initialize
 - onlyOwner
- createVesting
 - whenNotPaused
 - onlyPresale
- claimShare
 - nonReentrant
- pause
 - onlyOwner
- unpause
 - onlyOwner

JamonVault

- ✓ **depositTokens**
 - Ⓜ nonReentrant
- ✓ **stake**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓ **harvestToken**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓ **harvestAll**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓ **unStake**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓ **safeUnStake**
 - Ⓜ whenPaused
 - Ⓜ nonReentrant
- ✓ **updateBalances**
 - Ⓜ whenNotPaused
 - Ⓜ nonReentrant
- ✓ **updateTokenBalance**
 - Ⓜ onlyOwner
- ✓ **setTokenList**
 - Ⓜ onlyOwner
- ✓ **setApy**
 - Ⓜ onlyOwner
- ✓ **pause**
 - Ⓜ onlyOwner
- ✓ **unpause**
 - Ⓜ onlyOwner

JamonVesting

- ✓ **initialize**
 - Ⓜ onlyOwner
- ✓ **createVestingSchedule**
 - Ⓜ whenNotPaused
 - Ⓜ onlyBonus
- ✓ **release**
 - Ⓜ nonReentrant
 - Ⓜ whenNotPaused
 - Ⓜ onlyIfVestingSchedule
- ✓ **depositToVault**
 - Ⓜ nonReentrant
 - Ⓜ whenNotPaused
- ✓ **pause**
 - Ⓜ onlyOwner
- ✓ **unpause**
 - Ⓜ onlyOwner































Comments

- Deployer can set following state variables without any limitations
 - JamonShare
 - _totalSupply
 - JamonVesting
 - VestingSchedule amountTotal
 - Bonus
 - Proposal
 - startBlock_
 - endBlock
 - hardcap
 - FEEDS[token_].decimals
 - JamonShareVesting
 - Mint/SHATE_VESTING
- Deployer can enable/disable following state variables
 - JamonVesting
 - _paused
 - JamonSharePresale
 - listActive
 - WhitelistLP
 - Max4Wallet






























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/IJamonSharePresale.sol	—————	1	6	5	3	1	3	—————
	contracts/interfaces/IConversor.sol	—————	1	6	5	3	1	3	
	contracts/interfaces/IJamonVesting.sol	—————	1	11	5	3	1	5	—————
	contracts/interfaces/IJamonShareVesting.sol	—————	1	10	5	3	1	3	—————
	contracts/interfaces/IERC20MintBurn.sol	—————	1	11	7	4	1	9	—————
	contracts/interfaces/IJamonRouter.sol	—————	1	38	5	3	1	9	—————
	contracts/interfaces/IJamonPair.sol	—————	1	18	7	4	1	9	—————
	contracts/tokens/JamonShare.sol	1	—————	41	32	23	2	22	—————
	contracts/tokens/JamonV2.sol	1	—————	54	54	44	3	48	
	contracts/JamonVesting.sol	1	1	357	304	192	93	132	
	contracts/JamonShareVault.sol	1	—————	491	464	295	143	280	
	contracts/Conversor.sol	1	—————	305	290	210	68	142	
	contracts/JamonSharePresale.sol	1	—————	501	443	305	126	257	—————
	contracts/Bonus.sol	1	—————	338	315	196	96	151	
	contracts/JamonShareVesting.sol	1	—————	109	109	61	37	51	—————
	contracts/JamonGovernor.sol	1	—————	133	98	40	47	44	
	contracts/JamonVault.sol	1	—————	518	491	312	154	294	
	Totals	10	8	2947	2639	1701	776	1462	   

v1.1

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/interfaces/IJamonSharePresale.sol	————	1	6	5	3	1	3	————
	contracts/interfaces/IConversor.sol	————	1	6	5	3	1	3	
	contracts/interfaces/IJamonVesting.sol	————	1	11	5	3	1	5	————
	contracts/interfaces/IJamonShareVesting.sol	————	1	10	5	3	1	3	————
	contracts/interfaces/IERC20MintBurn.sol	————	1	11	7	4	1	9	————
	contracts/interfaces/IJamonRouter.sol	————	1	38	5	3	1	9	————
	contracts/interfaces/IJamonPair.sol	————	1	18	7	4	1	9	————
	contracts/tokens/JamonShare.sol	1	————	42	33	23	3	22	————
	contracts/tokens/JamonV2.sol	1	————	54	54	44	3	48	
	contracts/JamonVesting.sol	1	1	357	304	192	93	133	
	contracts/JamonShareVault.sol	1	————	519	492	318	148	291	
	contracts/Conversor.sol	1	————	353	338	250	76	173	
	contracts/JamonSharePresale.sol	1	————	543	485	332	141	280	————
	contracts/Bonus.sol	1	————	457	422	289	109	201	
	contracts/JamonShareVesting.sol	1	————	119	112	65	37	56	————
	contracts/JamonGovernor.sol	1	————	133	98	40	46	44	
	contracts/JamonVault.sol	1	————	534	507	323	159	306	
	Totals	10	8	3211	2884	1899	822	1595	  

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

High issues

No high issues

Medium issues

No medium issues

Low issues

Issue	File	Type	Line	Description
#1	JamonSharePreSale	Missing Zero Address Validation (missing-zero-check)	103-108	Check that the address is not zero
#2	JamonShareVault	Missing Zero Address Validation (missing-zero-check)	61-63	Check that the address is not zero
#3	JamonShareVesting	Missing Zero Address Validation (missing-zero-check)	50	Check that the address is not zero
#4	JamonShareVesting	Missing Events Access Control	59	Emit an event for critical parameter changes
#5	JamonVesting	Missing Events Access Control	76	Emit an event for critical parameter changes

Informational issues

Issue	File	Type	Line	Description
-------	------	------	------	-------------

#1	Convers or	Unused return values	290, 318, 316, 288	Ensure that all the return values of the function calls are used and handle both success and failure cases if needed by the business logic
#2	JamonS harePre sale	Unused return values	502, 506	Ensure that all the return values of the function calls are used and handle both success and failure cases if needed by the business logic
#3	JamonS hareVau lt	Unused return values	68, 69, 491, 493	Ensure that all the return values of the function calls are used and handle both success and failure cases if needed by the business logic
#4	JamonV ault	Unused return values	66, 494, 496	Ensure that all the return values of the function calls are used and handle both success and failure cases if needed by the business logic
#7	JamonS hareVes ting	Unused state variables	34	Remove unused state variables
#8	JamonS hare	NatSpec documentation missing	-	If you start to comment your code, also comment all other functions, variables etc.
#9	JamonV 2	NatSpec documentation missing	-	If you start to comment your code, also comment all other functions, variables etc.

Audit Comments

27. January 2022:

- [Read whole report for more information](#)

03. February 2022:

- [Read whole report for more information](#)



SWC Attacks

ID	Title	Relationships	Status
SW C-1 36	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	PASSED
SW C-1 35	Code With No Effects	CWE-1164: Irrelevant Code	PASSED
SW C-1 34	Message call with hardcoded gas amount	CWE-655: Improper Initialization	PASSED
SW C-1 33	Hash Collisions With Multiple Variable Length Arguments	CWE-294: Authentication Bypass by Capture-replay	PASSED
SW C-1 32	Unexpected Ether balance	CWE-667: Improper Locking	PASSED
SW C-1 31	Presence of unused variables	CWE-1164: Irrelevant Code	NOT PASSED
SW C-1 30	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	PASSED
SW C-1 29	Typographical Error	CWE-480: Use of Incorrect Operator	PASSED
SW C-1 28	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	PASSED

SW C-1 27	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SW C-1 25	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SW C-1 24	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SW C-1 23	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SW C-1 22	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SW C-1 21	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SW C-1 20	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	PASSED
SW C-11 9	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	PASSED
SW C-11 8	Incorrect Constructor Name	CWE-665: Improper Initialization	PASSED
SW C-11 7	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	PASSED

SW C-11 6	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SW C-11 5	Authorization through tx.origin	CWE-477: Use of Obsolete Function	PASSED
SW C-11 4	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	PASSED
SW C-11 3	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	PASSED
SW C-11 2	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SW C-11 1	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	PASSED
SW C-11 0	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	PASSED
SW C-1 09	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	PASSED
SW C-1 08	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED
SW C-1 07	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	PASSED
SW C-1 06	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	PASSED

SW C-1 05	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	PASSED
SW C-1 04	Unchecked Call Return Value	CWE-252: Unchecked Return Value	PASSED
SW C-1 03	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	PASSED
SW C-1 02	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	PASSED
SW C-1 01	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	PASSED
SW C-1 00	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED

The logo features the words "SolidProof" in a white, handwritten-style script. The "P" is large and stylized, with a long horizontal stroke that extends to the left. The background is a solid blue color with a faint, large shield emblem. The shield has a grid-like pattern on its right side and a solid blue area on its left side.

SolidProof

Blockchain Security | Smart Contract Audits | KYC

A small horizontal bar representing the German flag, with black, red, and gold stripes.

MADE IN GERMANY