SOLIDProof
Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC Development | Marketing

MADE IN GERMANY

Walker World

Audit

Security Assessment
07. March, 2023

For

WALKER WORLD

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 10
Risk Level 10
Capabilities 11
Inheritance Graph 13
CallGraph 14
Scope of Work/Verify Claims 15
Modifiers and public functions 24
Source Units in Scope 30
Critical issues 32
High issues 32
Medium issues 32
Low issues 32
Informational issues 33
Commented Code exist 34
Audit Comments 35
SWC Attacks 36

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	06. March 2023	• Layout project • Automated- /Manual-Security Testing • Summary

Network
Ethereum

Website
https://walkerworld.io/

Twitter
https://twitter.com/walkerworld_

YouTube
https://www.youtube.com/@walkerworld

Description

Walker World is an open world experience powered by Unreal Engine 5 by some of the most experienced and skilled AAA Directors, Artists and Developers in the Web 3 gaming industry. We are heavily focused on interoperability and giving value back to the player through digital asset ownership and in-game rewards.

Project Engagement

During the Date of 2 March 2023, **Walker World 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

WALKER WORLD

Contract Link

v1.0

https://github.com/x-continuumlabs-x/ww_soft_staking
Commit: e3d5890

Vulnerability & Risk Level

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

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

Auditing Strategy and Techniques Applied

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

Methodology

The auditing process follows a routine series of steps:

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

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

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

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

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

Imported packages:

Dependency / Import Path	Count
@openzeppelin/contracts/access/AccessControl.sol	3
@openzeppelin/contracts/access/Ownable.sol	15
@openzeppelin/contracts/security/ReentrancyGuard.sol	5
@openzeppelin/contracts/token/ERC1155/ERC1155.sol	2
@openzeppelin/contracts/token/ERC1155/IERC1155.sol	2
@openzeppelin/contracts/token/ERC1155/extensions/ERC1155Supply.sol	2
@openzeppelin/contracts/token/ERC1155/utils/ERC1155Holder.sol	1
@openzeppelin/contracts/token/ERC20/ERC20.sol	1
@openzeppelin/contracts/token/ERC20/IERC20.sol	2
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	1
@openzeppelin/contracts/token/ERC721/ERC721.sol	5
@openzeppelin/contracts/token/ERC721/IERC721.sol	5
@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol	3
@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol	2
@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol	4
@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol	1
@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol	3
@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol	3
@openzeppelin/contracts/token/ERC721/utils/ERC721Holder.sol	1
@openzeppelin/contracts/token/common/ERC2981.sol	1
@openzeppelin/contracts/utils/Address.sol	5
@openzeppelin/contracts/utils/Base64.sol	1
@openzeppelin/contracts/utils/Context.sol	3
@openzeppelin/contracts/utils/Counters.sol	3
@openzeppelin/contracts/utils/Strings.sol	6
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	3
@openzeppelin/contracts/utils/cryptography/MerkleProof.sol	4
@openzeppelin/contracts/utils/introspection/ERC165.sol	3
@openzeppelin/contracts/utils/math/SafeMath.sol	2
closedsea/src/OperatorFilterer.sol	1
erc721a/contracts/ERC721A.sol	2
erc721a/contracts/extensions/ERC721AQueryable.sol	1
erc721a/contracts/extensions/IERC721AQueryable.sol	1
hardhat/console.sol	2

Tested Contract Files

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

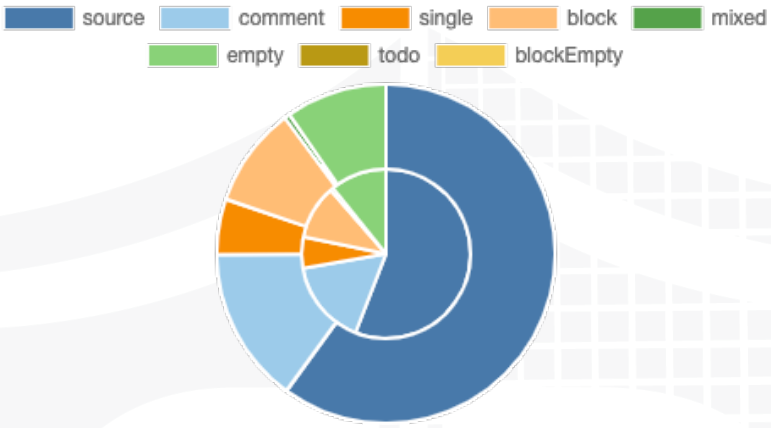
A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

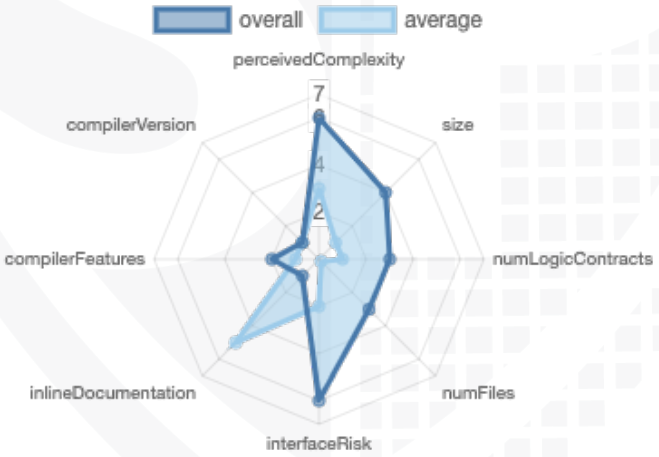
File Name	SHA-1 Hash
contracts/VaultInterface.sol	3199c098d2d6e66918737d488ddfa190a0c9e0f6
contracts/collections/CWcollection.sol	6f9a40b45fb6d8c608afd82c163a0d51cd084404
contracts/collections/CWWcollection.sol	9adee2a5d7f6d036aa5b8b8b79d3b03cd43d0b62
contracts/collections/CWF.sol	18839c2103bc251bae0e8e0166c42ad0221e3674
contracts/collections/CWBBcollection.sol	279d6a63b0a260295f381ca8a87f3b340e873b33
contracts/WWStakingManager.sol	28dc913cfa39c392d7a6508fd258bab91b0ff96a
contracts/WWStakingVault.sol	d1c66fb3f9ebf845db92eaa8fd2c941c3c441727
contracts/vault_collections/XoneNFT.sol	0bf9ea544af36fd0d1e6e2654f1e6737eef01779
contracts/vault_collections/ChefSaleManager.sol	6ccb658c4e87867afbc3cd2c733100bcfac28fe5
contracts/vault*collections/1155*hat.sol	29b03db91bbd52bb30444a8bd99d3182839e4ae8
contracts/vault_collections/ERC721A-OGREX.sol	22cfc4642c4267c5fc9352a7c2bd64010d08f2ea
contracts/vault_collections/IASMBrainGenII.sol	d012eb306f84053097184698b648a954c47e970b
contracts/vault_collections/OGRex.sol	6438398d43347496ad9b18c41ae886d632ff9c56
contracts/vault*collections/721*schmoodles.sol	1dccb306d6cfd4b787efcc2a16b40f0cd6118a8b
contracts/vault_collections/Util.sol	e202a7e851e78da989129b59b7a25c4e6ec62235
contracts/vault*collections/721A*cats.sol	26325dedb78dafe5487c66d86f3e028bc7cedc95
contracts/vault_collections/ERC721A-CHEF.sol	982b6c94ef68fe8972622e38a0b271572203c620
contracts/vault_collections/AccessTokenContract.sol	d042bbc51dd5b342eb369bd9fdc6a65d3ab2dcc0
contracts/vault_collections/Base58.sol	fda47a464a33c0c945f1290aad8f1e225db54ab2
contracts/vault_collections/ASMBrainGenII.sol	89b5c6cdc14d81d70960782c16c45152a73e94a2
contracts/vault*collections/20*coin.sol	b17d3f69507712ad34b597f19845b9c605b7ac40
contracts/vault_collections/ERC721A-SEEKER.sol	982b6c94ef68fe8972622e38a0b271572203c620
contracts/vault_collections/ChefAvatar.sol	570c322dd41c4aa0d68f1e9226c4db664065204c
contracts/vault_collections/Seeker.sol	a7eb695674ad9fe670ab6391bdebda73b26cc57e

Metrics

Source Lines
v1.0



Risk Level
v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	22	0	5	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	254	18

Version	External	Internal	Private	Pure	View
1.0	123	291	15	11	127

State Variables

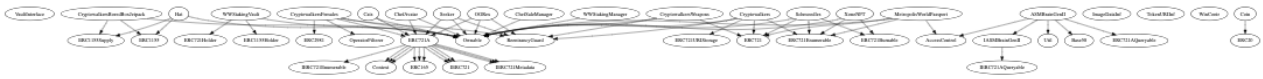
Version	Total	Public
1.0	198	85

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	`0.8.17` `^0.8.0` `^0.8.7` `^0.8.9` `^0.8.13` `^0.8.6` `^0.8.4` `^0.8.1`		`yes`	`yes` (3 asm blocks)	

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. Is contract an upgradeable
2. Overall checkup (Smart Contract Security)

Is contract an upgradeable

Name	
Is contract an upgradeable?	No

Write functions of contract v1.0

CRYPTOWALKERSBOREDBOXJETPACK

- mint
- renounceOwnership
- safeBatchTransferFrom
- safeTransferFrom
- setApprovalForAll
- setURI
- transferOwnership

CRYPTOWALKERS

- addUsersToWhitelistByIndex
- addUserToWhitelistByIndex
- approve
- changeWalkerDetails
- mintReserve
- mintWalkers
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setDetailsPricing
- setMintPricing
- setStateToPublic
- setStateToSetup
- setStateToWhitelist
- setTokenURI
- transferFrom
- transferOwnership
- withdrawAllEth

CRYPTOWALKERSFEMALES

- mint
- approve
- decreaseSupply
- mintReserve
- mintToVault
- presaleMint
- renounceOwnership
- repeatRegistration
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setBaseURI
- setOperatorFilteringEnabled
- setRootHash
- setStateToAdmin
- setStateToEarlyBird
- setStateToPaused
- setStateToPresale
- setStateToPublic
- setVault
- transferFrom
- transferOwnership
- withdrawAll

CRYPTOWALKERSWEAPONS

- adminMint
- approve
- disableFreeMint
- enableFreeMint
- freeMint
- msMint
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setStateToAdmin
- setStateToPaused
- setStateToPublic
- transferFrom
- transferOwnership
- updateBaseURI
- withdrawAll

WWSTAKINGMANAGER

- addPoints
- adminAddPoints
- adminUnstake
- claimItems
- overridePremiumIds
- removePoints
- setERC20Collection
- renounceOwnership
- setCollectionIds
- setERC1155Collection
- setERC721Collection
- setLeagueHurdles
- setPointsAdmin
- setPointsMultipliers
- setRootHash
- setSquadMultiplier
- setStateToArchived
- setStateToPublic
- setVault
- stakeMultiple
- transferOwnership
- unStake

WWSTAKINGVAULT

- addERC1155ToVault
- addERC20ToVault
- addERC721ToVault
- onERC1155BatchReceived
- onERC1155Received
- onERC721Received
- setAdmin
- setPhysicalStock
- transferItems
- transferOwnership
- renounceOwnership
- setCost
- setVaultCollection
- withdrawTokens

XONENFT

- airdrop
- airdropGiftMode
- approve
- burn
- createPlan
- mintTeam
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setBaseURI
- setWhiteLists
- transferFrom
- transferOwnership
- updateCurrentPlan
- updatePlanState
- withdrawETH

SEEKER

- approve
- discountedMint
- forwardEther
- mint
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setStateToFinished
- setStateToPublicSale
- setStateToSetup
- setTokenURI
- transferFrom
- transferOwnership
- updateBeneficiaryWallet
- updateSigner
- withdrawAll
- withdrawAllViaCall

OGREX

- approve
- changeOGRexDetails
- forwardEth
- mintOGRex
- mintReserve
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setEndorser
- setMintPricing
- setRecipientWallet
- setStateToClosed
- setStateToPresale
- setStateToPublic
- setStateToSetup
- setTokenURI
- transferFrom
- transferOwnership
- withdrawAllEth

ERC721ACHEF/-OGREX/-SEEKER

- approve
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- transferFrom

CHEFAVATAR

- approve
- mint
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setBaseTokenURI
- setChefSaleManager
- transferFrom
- transferOwnership

CHEFSALEMANAGER

- configureDutch
- presaleBuy
- publicBuy
- renounceOwnership
- setMerkleRoot
- setPresaleConfig
- setPrices
- setPublicConfig
- setPublicSaleMaxPurchaseQuantity
- setPublicSalePricingModel
- setTreasury
- transferOwnership

METROPOLISWORLDPASSPORT

- approve
- bulkMint
- freeMint
- grantRole
- renounceOwnership
- renounceRole
- revokeRole
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setImageContract
- setMaxAllowed
- setPrice
- setWLContractAddress
- transferFrom
- transferOwnership
- userFreeMint

ASMBRAINGENII

- addAdmin
- addMinter
- approve
- grantRole
- mint
- removeAdmin
- removeMinter
- renounceRole
- revokeRole
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- transferFrom
- updateBaseURI

SCHMOODLES

- approve
- burn
- renounceOwnership
- safeMint
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- transferFrom
- transferOwnership

HAT

- mint
- renounceOwnership
- safeBatchTransferFrom
- safeTransferFrom
- setApprovalForAll
- setURI
- transferOwnership

CATS

- approve
- mint
- renounceOwnership
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- transferFrom
- transferOwnership

COIN

- approve
- decreaseAllowance
- increaseAllowance
- transfer
- transferFrom

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

```
setVault
    onlyOwner
setERC721Collection
    onlyOwner
setERC1155Collection
    onlyOwner
setERC20Collection
    onlyOwner
setCollectionIds
    onlyOwner
setPointsMultipliers
    onlyOwner
setSquadMultiplier
    onlyOwner
setLeagueHurdles
    onlyOwner
setRootHash
    onlyOwner
overridePremiumIds
    onlyOwner
setPointsAdmin
    onlyOwner
addPoints
adminAddPoints
    onlyOwner
removePoints
    onlyOwner
stakeMultiple
unStake
adminUnstake
    onlyOwner
claimItems
setStateToPublic
    onlyOwner
setStateToArchived
    onlyOwner
```

Comments

- *Deployer can set following state variables without any limitations*
  - WWStakingManager
    - _maxAddablePoints
    - premiumLevel
    - secondaryLevel
    - squadMultiplier
    - premiumMultipliers
    - secondaryMultipliers
    - collectionIds
  - WWStakingVault

        - cost
        - physicalStock
    - CWBBcollection
    - CWcollection
        - WALKER_PRICE
        - UPDATE_DETAILS_PRICE
    - CryptowalkersFemales
    - AccessTokenContract
        - _maxAllowedPerWallet
        - _navPrice
- ChefSaleManager
    - presalePrice
    - publicFixedPrice
    - presaleStart
    - presaleLength
    - publicStart
    - publicSaleMaxPurchaseQuantity
    - publicSalePricingModel
    - dutchStartPrice
    - dutchEndPrice
    - dutchPriceStepDrecrease
    - dutchStartTime
    - dutchStep
- OGRex
    - OGREX_PRICE

- *Deployer can enable/disable following state variables*
    - WWStakingManager
        - activeState
            - Can be set to
                - public
                - archived
    - CWBBcollection
    - CWcollection
    - CryptowalkersFemales
        - operatorFilteringEnabled
    - CryptowalkersWeapons
        - freeMintActive
- XoneNFT
    - whitelists

- *Deployer can set following addresses/String*
    - WWStakingManager
        - pointsAdmin
        - rootHash

        - vault
    - WWStakingVault
        - admin
    - CWBBcollection
    - CWcollection
        - _tokenUriBase
    - CryptowalkersFemales
        - vaultAddress
        - rootHash
        - tokenBaseURI
    - CryptowalkersWeapons
        - _baseTokenURI
    - Hat
        - _uri
    - ASMBrainGenII
        - baseURI
- ChefAvatar
    - _baseTokenURI
    - saleManager
- ChefSaleManager
    - merkleRoot
    - treasury
- OGRex
    - baseURIString
    - endorser
    - state
        - Can be set to
            - setup
            - Presale
            - Public
            - Closed
- Seeker
    - _beneficiaryWallet
    - _signer
    - _tokenUriBase
    - Set state to
        - Setup
        - publicSale
        - Finished
- XoneNFT
    - baseTokenURI

- *Existing Modifiers*
    - WWStakingManager

        - onlyOwner
    - CWBBcollection
    - CryptowalkersFemales

- WWStakingManager
    - Owner is able to
        - Unstake for staker without allowance
        - Add/Remove points from staker
            - It can be done without limitations
        - Add a league uint
        - Add new
            - ERC20 collection
            - ERC1155 collection
            - ERC721 collection
- WWStakingVault
    - Owner is able to
        - withdraw all assets
        - Set and add details for a new collection being added to the vault
        - Add ERC721 NFTs to vault for specific collection
        - Add ERC1155 NFTs to vault for specific collection
        - Add ERC20 NFTs to vault for specific collection
- CWBBcollection
    - Owner is able to
        - Mint new tokens to arbitrary addresses
- CWcollection
    - Owner is able to
        - Withdraw all eth from contract
        - Set state to setup whitelist public
        - Add user to whitelist by index
        - Mint new tokens (max up to 10.000)
        - Mint new reserves (max up to 400)
- CryptowalkersFemales
    - Owner is able to
        - Mint reserves (max up to 200)
        - Mint to vault (max up to 1625)
        - Approval for all for an arbitrary address
        - Withdraw all eth from contract
        - Set activeState to
            - Paused
            - Public
            - Presale
            - earlybird
            - admin

        - Decrease max supply
- CryptowalkersWeapons
    - Owner is able to
        - Withdraw all eth from contract
        - Set activeState to
            - Paused
            - Public
            - Admin
        - Mint new tokens to himselfs address (max up to 502)
- Schmoodles
    - Owner is able to
        - Mint nfts without limitations
- Cats
    - Owner is able to
        - Mint tokens without limitations
- Hat
    - Owner is able to
        - Mint tokens without limitations
- AccessTokenContract
        - Address with updater_Role
        - can freeMint tokens for arbitrary addresses
        - Can set
            - imageContract address
            - WIN_Contract/WinContract
            - WL_CONTRACT/TURI_CONTRACT/TuriContract/_paymentSplit
- ASMBrainGenII
    - Only ADMIN_ROLE can
        - revoke/add Admin Role
        - revoke/add Minter Role
    - Only MINTER_ROLE can
        - Mint new tokens
- ChefAvatar
    - SaleManager can mint new tokens without limitations
- OGRex
    - Owner is able to
        - Mint new tokens with a cap of 7777 and 807
        - Withdraw contract balance to an arbitrary address
        - Set the minting price to any arbitrary amount
- Seeker
    - Anyone is able to mint/discountedMint new tokens without paying something
    - Owner is able to
        - Withdraw contract balance to an arbitrary address
- XoneNFT

- Owner is able to
  - Create new plans
  - Update plans
  - Mint for team
  - Withdraw etc to an arbitrary address
  - Update current plan
  - Airdrop to an arbitrary address
  - Airdrop gift to an arbitrary address

THe sales manager address can mint tokens in the ChefAvatar contract
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/VaultInterface.sol	————	1	13	6	3	1	3	————
📝	contracts/collections/CWcollection.sol	1	————	287	247	204	10	132	💰
📝	contracts/collections/CWWcollection.sol	1	————	202	180	158	2	115	💰🧱
📝	contracts/collections/CWF.sol	1	————	323	253	212	9	171	💰🧱
📝	contracts/collections/CWBBcollection.sol	1	————	43	32	24	2	22	————
📝	contracts/WWStakingManager.sol	1	————	801	662	550	50	320	🧱
📝	contracts/WWStakingVault.sol	1	————	486	429	345	54	169	🧱
📝	contracts/vault_collections/XoneNFT.sol	1	————	353	297	245	10	176	————
📝	contracts/vault_collections/ChefSaleManager.sol	1	————	343	317	220	47	103	💰🧱
📝	contracts/vault*collections/1155*hat.sol	1	————	36	29	21	2	20	————
📝	contracts/vault_collections/ERC721A-OGREX.sol	1	————	573	493	279	150	199	🖥♻
🔍	contracts/vault_collections/IASMBrainGenII.sol	————	1	73	22	11	42	19	————
📝	contracts/vault_collections/OGRex.sol	1	————	305	284	154	103	107	💰💠🧱
📝	contracts/vault*collections/721*schmoodles.sol	1	————	46	36	27	2	26	————
📝	contracts/vault_collections/Util.sol	1	————	53	53	30	18	26	🧱
📝	contracts/vault*collections/721A*cats.sol	1	————	25	25	18	2	11	————
📝	contracts/vault_collections/ERC721A-CHEF.sol	1	————	669	594	296	225	196	🖥♻Σ
📝🔍	contracts/vault_collections/AccessTokenContract.sol	1	3	295	224	167	40	141	💰🧱
📝	contracts/vault_collections/Base58.sol	1	————	61	57	39	14	60	————
📝	contracts/vault_collections/ASMBrainGenII.sol	1	————	168	151	76	57	99	————
📝	contracts/vault*collections/20*coin.sol	1	————	10	10	7	1	5	————
📝	contracts/vault_collections/ERC721A-SEEKER.sol	1	————	669	594	296	225	196	🖥♻Σ
📝	contracts/vault_collections/ChefAvatar.sol	1	————	115	110	63	22	40	————
📝	contracts/vault_collections/Seeker.sol	1	————	277	250	184	15	125	💰💠🧱
📝🔍	**Totals**	**22**	**5**	**6226**	**5355**	**3629**	**1103**	**2481**	🖥💰💠🧱♻Σ

Legend

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

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

Audit Results

Critical issues

No critical issues

High issues

No high issues

Medium issues

No medium issues

Low issues

Issue	File	Type	Line	Description
#1	CWF.sol	Contract doesn't import npm packages from source (like OpenZeppelin etc.)	12	We recommend to import all packages from npm directly without flatten the contract. Functions could be modified or can be susceptible to vulnerabilities
#2	All(except collections)	A floating pragma is set	All	The current pragma Solidity directives are ,"^0.8.13", "^0.8.7", and "^0.8.9".
#3	XoneNFT.sol	Missing Zero Address Validation (missing-zero-check)	73	Check that the address is not zero
#4	CWcollection.sol	Missing Events Arithmetic	100-116, 251-259	Emit an event for critical parameter changes
#5	CWF.sol / CWW.sol	Missing Events Arithmetic	All	Emit an event for critical parameter changes
#6	WWStakingManager	Wrong property was used	222	Mody the premiumLevel to secondaryLevel

#7	OGRex.sol	Wrong Implementation	255	The changeOGRexDetails function has no impact on the state of the contract because the function doesn't have the necessary logic to change the details of a Token.
#8	CWcollection.sol	Wrong Implementation	228	The changeWalkerDetails function has no impact on the details of the token because the function doesn't have the necessary logic to change the details of a Token. It only sets the updated status to true, but doesn't make any changes to the Name and Description of the token.
#9	CWcollection.sol	Local Variables shadowing	233	Rename the variables that shadows other component in of the inherited contract
#10	WWStakingVault.sol	Weak Randomisation	274	We recommend using Off chain randomisation

Informational issues

Issue	File	Type	Line	Description
#1	CWBBcollection	State variables that could be declared constant (constable-states)	11, 12	Add the `constant` attributes to state variables that never change
#2	CWcollection.sol	Functions that are not used	263	Remove unused functions. Before removing check the function, it could be possible, that you forget to implement it into the contract

#3	WWStakingVault	Misspelling	See description	Change following words: - colleciton L88 - Overwitting L90 - Make sure to change it everywhere else as well.
#4	XoneNFT.sol	Error message is missing	179, 180	Provide an error message for require statement
#5	All	NatSpec documentation missing	All	If you started to comment your code, also comment all other functions, variables etc.
#6	WWStakingVault	withdrawTokens check balance	441, 460, 477	We recommend you to check the balance in the for loop instead outside of it because in the loop the item (ERC721, ERC1155, etc.) will be transferred to the recipient
#7	CryptowalkersBoredBoxJetpack	Modify URL	14	In the contract it Is still "https://example.com/" as an example. Modify it
#8	CWcollection	Modify URL	70	In the contract it Is still "https://example.com/" as an example. Modify it
#9	721_schmoodles	Modify URL	18	In the contract it Is still "https://example.com/" as an example. Modify it
#10	AccessTokenContract	Free mintable	107	The freeMintable state variable will be always 0.

Commented Code exist

There are some instances of code being commented out in the following files that should be removed:

File	Line	Comment
ChefAvatar.sol	100-114	N/A
CWcollection.sol	210-213	N/A
Seeker.sol	138, 140	N/A

Recommendation

Remove the commented code, or address them properly.

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.

7. March 2023:

- There is still an owner (Owner still has not renounced ownership)
- Read whole report and modifiers section for more information

SWC Attacks

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

SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	PASSED
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	NOT PASSED
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	PASSED
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	PASSED

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

SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	PASSED
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	PASSED
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	NOT PASSED
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	PASSED
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	PASSED
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED

SolidProof_io

@solidproof_io

Solid Proofed

Blockchain Security | Smart Contract Audits | KYC Development | Marketing

MADE IN GERMANY