SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

Audit

Security Assessment

20. December, 2021

For

vEMPIRE

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	20. December 2021	• Layout project • Automated- /Manual-Security Testing • Summary

Network
Ethereum (ERC20)

Website
https://v-empire.digital/

Telegram
https://t.me/vEmpirediscussion

Twitter
https://twitter.com/vEmpiredigital

Facebook
https://www.facebook.com/vEmpireDDAO

Instagram
https://www.instagram.com/vempire.digital/

Reddit
https://www.reddit.com/r/vEmpireDDAO/

Medium
https://medium.com/@v-empire.digital

LinkedIn
https://www.linkedin.com/company/vempire-ddao-ltd/

Youtube
https://www.youtube.com/channel/UCjhhTUTgN2xW7IAAXSxvHrw

Description

The vEmpire DDAO distributes value generated by a basket of pools and LP services to stakeholders. The DDAO functions as a cooperative, whereby stakeholders earn vEmpire's token (VEMP) for providing collateral and, via a staking mechanism, receive a share of the fee revenues generated by supported DeFi services, pools, NFTs and any fees generated from the DDAOs contributions on the platform or in any metaverse.

The VEMP work token effectively encapsulates the intrinsic value of the VEMP services basket. The VEMP token can be staked into xVEMP to grant pro-rata governance rights over all operation concerns of the DeFi services' provision. Income generated for the Empire will be gifted to xVEMP holders. Staking derivatives will also be enabled via locked pools on top of the supported DeFi protocols.

Project Engagement

During the 7th of December 2021, **vEmpire 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

ERCXX

vEMPIRE

Contract Link

v1.0

TBA

Github: https://github.com/v-Empire/vEmpire
Commit: dab1d2965e5466c4523e95a9acd715aad7c8d593

Testnetwork

- Game
    - https://rinkeby.etherscan.io/address/0x174D0631d77795a99F33B5fc7874F94f7324C704
- Battle
    - https://rinkeby.etherscan.io/address/0xa457bcb19083cfbd969e065c0435393fb62b0c02
- xsVEMP
    - https://rinkeby.etherscan.io/address/0xb3933d99c61f58c404d93947b37e5f1be35f140b
- xVEMPBEP20Token
    - https://rinkeby.etherscan.io/address/0xa97ea487d483c82c5a8e3bbd70590bf21b77bd34
- VempDao
    - https://rinkeby.etherscan.io/address/0xc6773d7b7458f70ae6585c45bca3d076e8537caf
- ProxyAdmin
    - https://rinkeby.etherscan.io/address/0x8bcbf931524b678dc7451a825791a3159fd46900
- Airdrop
    - https://rinkeby.etherscan.io/address/0x88D167e3BC30da9df107841a51c0DA4981A4C634
- MasterChefBEP20Vemp
    - https://rinkeby.etherscan.io/address/0x2A06DaE6c509598a0F52C205E7B414e86aFc83c7

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:

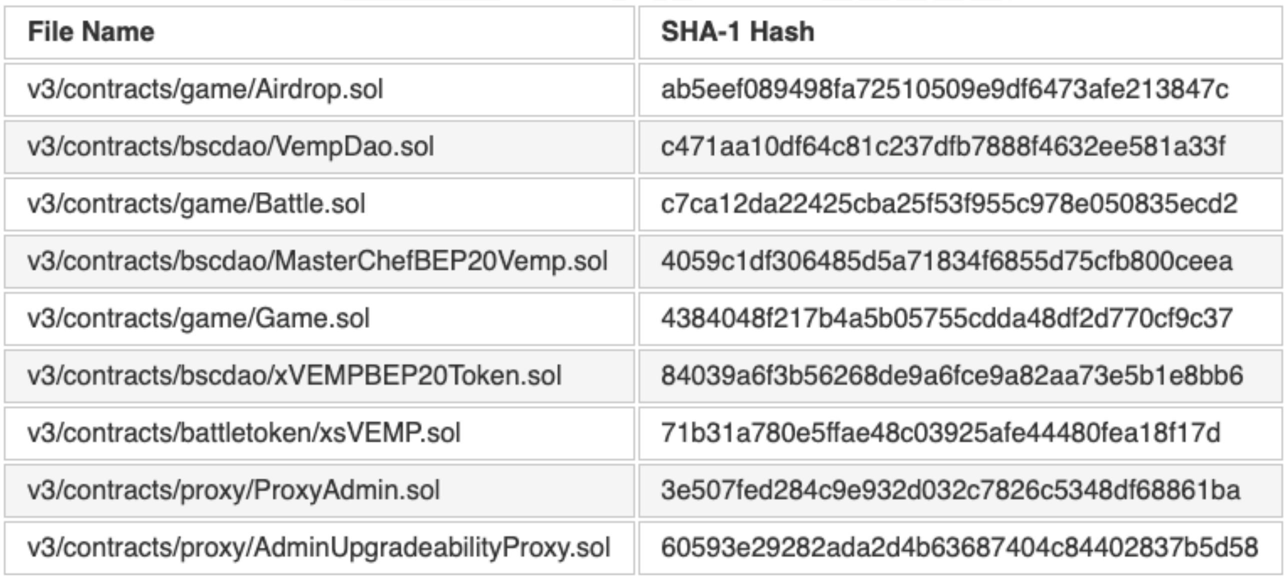
Dependency / Import Path	Count
@chainlink/contracts/src/v0.6/VRFConsumerBase.sol	1
@openzeppelin/contracts/GSN/Context.sol	2
@openzeppelin/contracts/access/Ownable.sol	2
@openzeppelin/contracts/math/SafeMath.sol	6
@openzeppelin/contracts/token/ERC20/ERC20Burnable.sol	1
@openzeppelin/contracts/token/ERC20/IERC20.sol	3
@openzeppelin/contracts/utils/Address.sol	2
@openzeppelin/contracts/utils/Context.sol	1

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

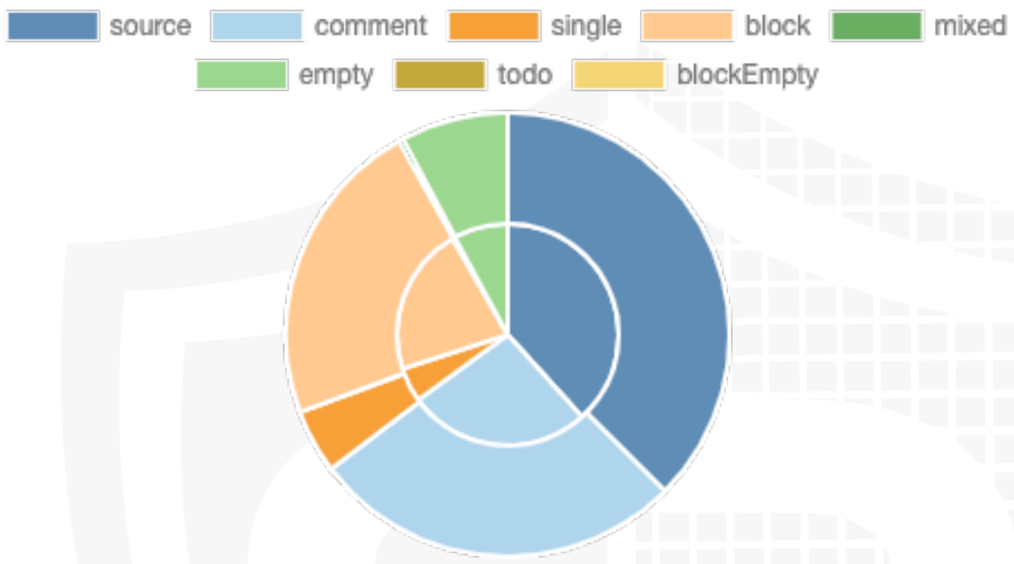


File Name	SHA-1 Hash
v3/contracts/game/Airdrop.sol	ab5eef089498fa72510509e9df6473afe213847c
v3/contracts/bscdao/VempDao.sol	c471aa10df64c81c237dfb7888f4632ee581a33f
v3/contracts/game/Battle.sol	c7ca12da22425cba25f53f955c978e050835ecd2
v3/contracts/bscdao/MasterChefBEP20Vemp.sol	4059c1df306485d5a71834f6855d75cfb800ceea
v3/contracts/game/Game.sol	4384048f217b4a5b05755cdda48df2d770cf9c37
v3/contracts/bscdao/xVEMPBEP20Token.sol	84039a6f3b56268de9a6fce9a82aa73e5b1e8bb6
v3/contracts/battletoken/xsVEMP.sol	71b31a780e5ffae48c03925afe44480fea18f17d
v3/contracts/proxy/ProxyAdmin.sol	3e507fed284c9e932d032c7826c5348df68861ba
v3/contracts/proxy/AdminUpgradeabilityProxy.sol	60593e29282ada2d4b63687404c84402837b5d58

Metrics

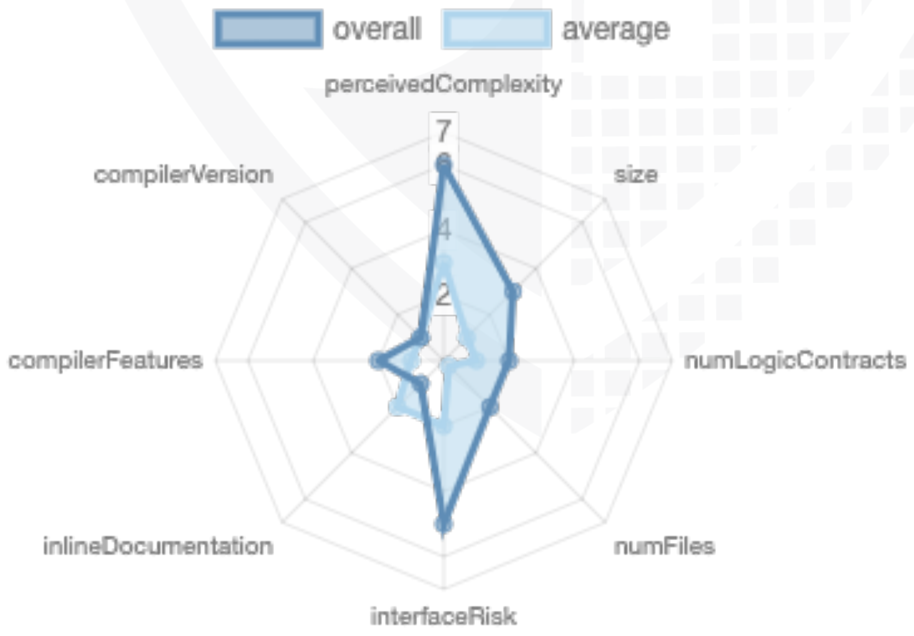
Source Lines

v1.0



Risk Level

v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	13	2	2	4

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	81	11

Version	External	Internal	Private	Pure	View
1.0	22	130	4	1	21

State Variables

Version	Total	Public
1.0	39	31

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	`=0.6.12`	`ABIEncoderV2`	`yes`	`yes` (14 asm blocks)	

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

Scope of Work

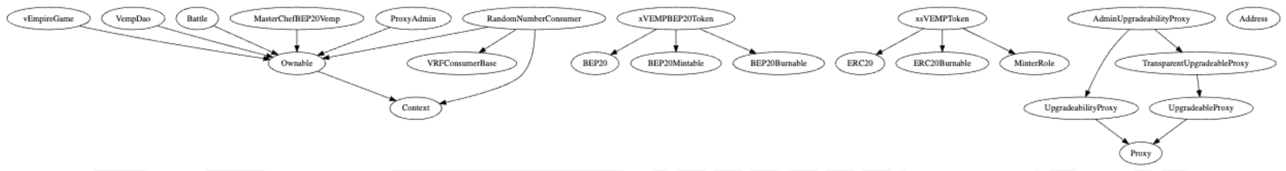
The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

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
✓	✓

Game

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	–	–	–

Battle

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	–	–	–

xsVemp

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	✓	✓	✓

xVEMPBEP20Token

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	✓	✓	✓

VempDao

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	–	–	–

ProxyAdmin

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	–	–	–

AdminUpgradeabilityProxy

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	–	–	–

Airdrop

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	–	–	–

MasterChefBEP20Vemp

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	–	–	–

Deployer cannot mint any new tokens

File	Name	Exist	Tested	Verified
Game	Deployer cannot mint	–	–	–
Battle	Deployer cannot mint	–	–	–
xsVEMP	Deployer cannot mint	✓	✓	✗
xVEMPBEP20Token	Deployer cannot mint	✓	✓	✗
VempDao	Deployer cannot mint	–	–	–
ProxyAdmin	Deployer cannot mint	–	–	–
AdminUpgradeability Proxy	Deployer cannot mint	–	–	–
Airdrop	Deployer cannot mint	–	–	–
MasterChefBEP20Ve mp	Deployer cannot mint	–	–	–

Max / Total Supply:

- xsVEMP
    - onlyMinter can mint and can be added by the owner
- xVEMPBEP20Token
    - onlyMinter can mint and can be added by the owner

Deployer cannot burn or lock user funds

File	Name	Exist	Tested	Verified
Game	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
Battle	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
xsVEMP	Deployer cannot lock	✓	✓	✓
	Deployer cannot burn	✓	✓	✓
xVEMPBEP20Token	Deployer cannot lock	✓	✓	✓
	Deployer cannot burn	✓	✓	✓
VempDao	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
ProxyAdmin	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
AdminUpgradeabilityProxy	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
Airdrop	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
MasterChefBEP20	Deployer cannot lock	–	–	–

Vemp	Deployer cannot burn	–	–	–

Deployer cannot pause the contract

File	Name	Exist	Tested	Verified
Game	cannot pause	–	–	–
Battle	cannot pause	–	–	–
xsVEMP	cannot pause	–	–	–
xVEMPBEP20Token	cannot pause	✓	✓	✗
VempDao	cannot pause	–	–	–
ProxyAdmin	cannot pause	–	–	–
AdminUpgradeabilityProxy	cannot pause	–	–	–
Airdrop	cannot pause	–	–	–
MasterChefBEP20Vemp	cannot pause	–	–	–

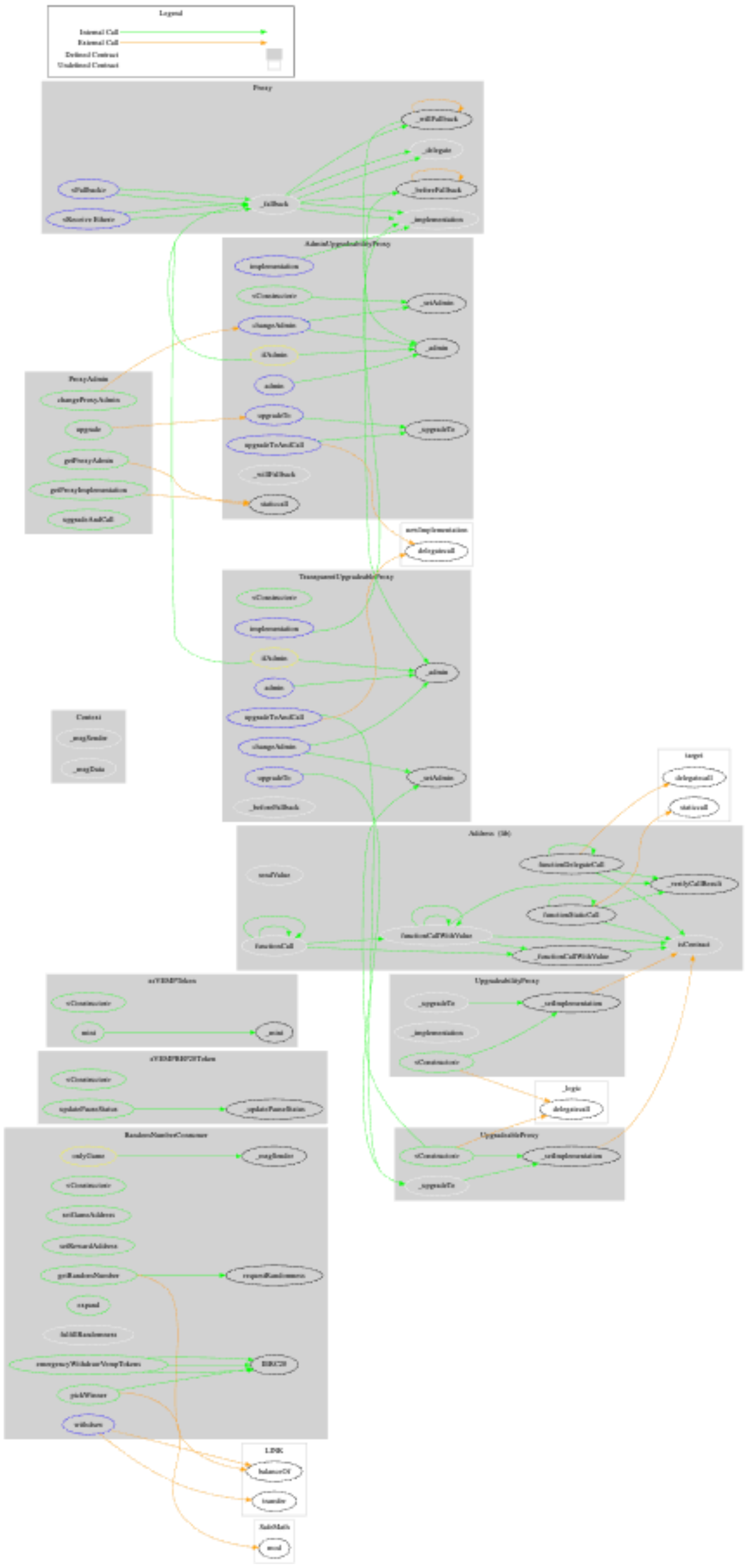
Overall checkup (Smart Contract Security)

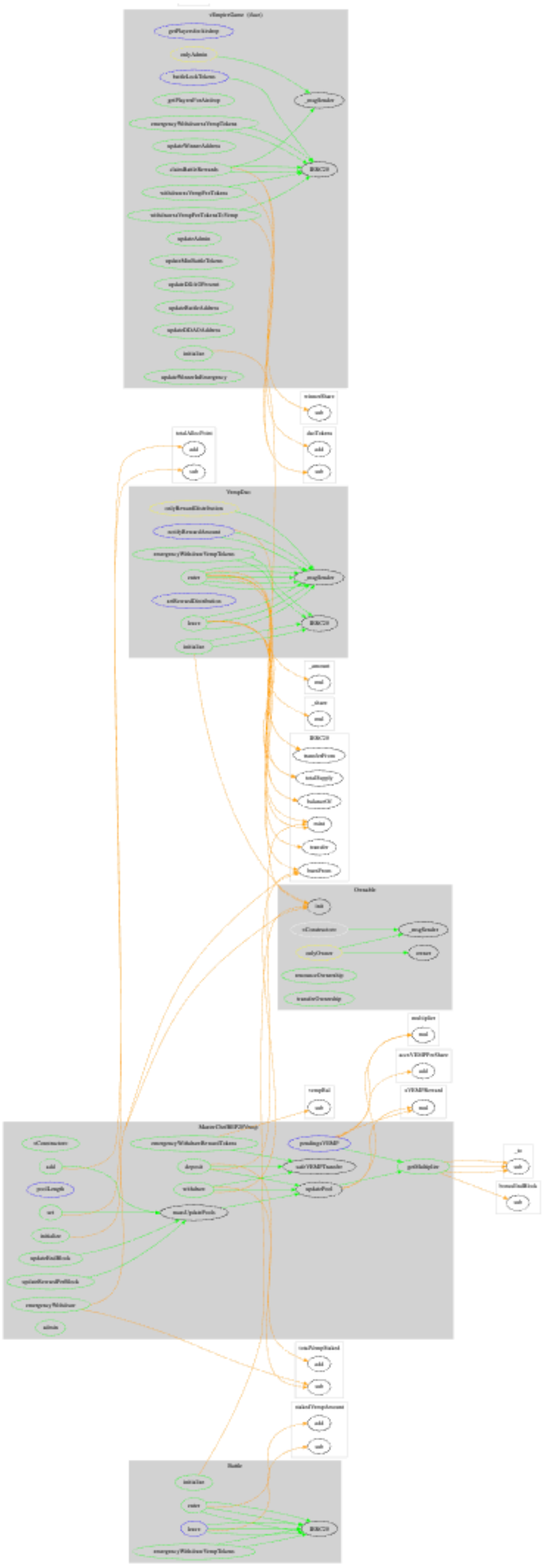
Tested	Verified
✓	✓

Legend

Attribute	Symbol
Verfified / Checked	✓
Partly Verified	⚑
Unverified / Not checked	✗
Not available	–

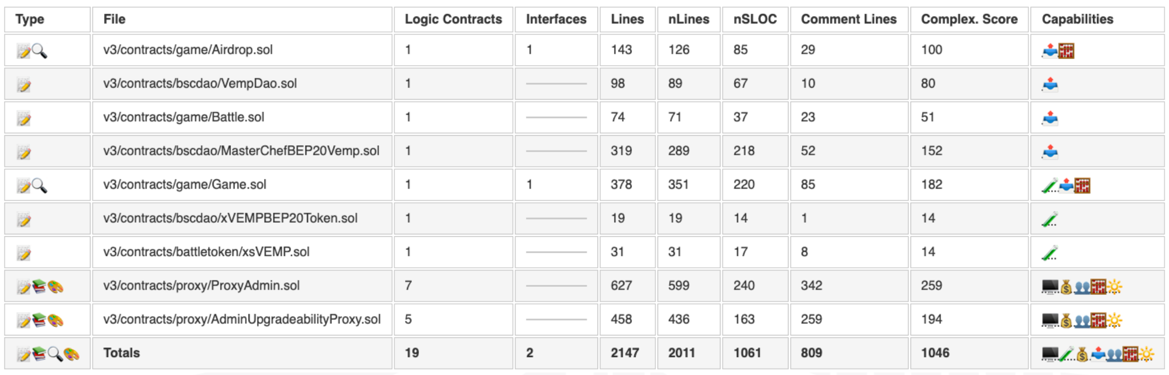
CallGraph
v1.0




vEmpireGame (iface)
getPlayersForAirdrop
onlyAdmin
battleLockTokens
getPlayersForAirdrop
emergencyWithdrawVempTokens
updateWinnerAddress
claimBattleRewards
withdrawxVempForTokens
withdrawxVempForTokensToVemp
updateAdmin
updateMinBattleTokens
updateDDAOPercent
updateBattleAddress
updateDDAOAddress
initialize
updateWinnerInEmergency
_msgSender
IERC20
winnerShare
sub
totalAllocPoint
add
sub
dscTokens
add
sub
VempDao
onlyRewardDistribution
notifyRewardAmount
emergencyWithdrawVempTokens
enter
setRewardDistribution
leave
initialize
_msgSender
IERC20
_amount
mul
_share
mul
IERC20
transferFrom
totalSupply
balanceOf
mint
transfer
burnFrom
Ownable
init
oConstructors
_msgSender
onlyOwner
owner
renounceOwnership
transferOwnership
multiplier
mul
accVEMPPerShare
add
vempBal
sub
xVEMPReward
mul
MasterChefBEP20Vemp
oConstructors
emergencyWithdrawRewardTokens
pendingxVEMP
add
deposit
safeVEMPTransfer
getMultiplier
poolLength
withdraw
updatePool
set
massUpdatePools
initialize
updateEndBlock
updateRewardPerBlock
emergencyWithdraw
admin
_to
sub
bonusEndBlock
sub
totalVempStaked
add
sub
stakedVempAmount
add
sub
Battle
initialize
enter
leave
emergencyWithdrawVempTokens
IERC20

Source Units in Scope

v1.0



Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
📝🔍	v3/contracts/game/Airdrop.sol	1	1	143	126	85	29	100	📤🧮
📝	v3/contracts/bscdao/VempDao.sol	1	—	98	89	67	10	80	📤
📝	v3/contracts/game/Battle.sol	1	—	74	71	37	23	51	📤
📝	v3/contracts/bscdao/MasterChefBEP20Vemp.sol	1	—	319	289	218	52	152	📤
📝🔍	v3/contracts/game/Game.sol	1	1	378	351	220	85	182	🖊📤🧮
📝	v3/contracts/bscdao/xVEMPBEP20Token.sol	1	—	19	19	14	1	14	🖊
📝	v3/contracts/battletoken/xsVEMP.sol	1	—	31	31	17	8	14	🖊
📝📚🎨	v3/contracts/proxy/ProxyAdmin.sol	7	—	627	599	240	342	259	🖥💰👥🧮☀️
📝📚🎨	v3/contracts/proxy/AdminUpgradeabilityProxy.sol	5	—	458	436	163	259	194	🖥💰👥🧮☀️
📝📚🔍🎨	**Totals**	**19**	**2**	**2147**	**2011**	**1061**	**809**	**1046**	🖥🖊💰📤👥🧮☀️

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
- no medium issues found -

Low issues
- no low issues found -

Informational issues
- no informational issues found -

vEmpire Game Test Results

Compiling successful
Artifacts written to build/contracts
Compiled successfully using:
- solc: 0.6.12+commit.27d51765.Emscripten.clang

Note: Duplicate contract names found for Battle.

Deployment
All contracts were deployed on local running blockchain node. No excessive gas usages were detected.

Unit testing
167 passing (5m)
8 failing

For details, please look at the separate testing protocol.

Conclusion
All problems found were discussed with the project managers and fixed immediately.
The contract is safe to deploy and basic logic errors were not detected.
The code is written to the best standard and sufficiently commented.
Known risks were checked by the auditor and the code was scanned for other vulnerabilities as well.

All unit tests make sense and have also been checked. Logic errors could not be found here either.

Disclaimer
We have checked and verified the code to the best of our knowledge. However, deeper logic errors cannot be excluded and Solidproof.io cannot be held liable for any damage that may occur.

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	**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	**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**

Solid Proofed

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY