



SOLIDProof
Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

Audit

Security Assessment
12. December, 2021

For



Disclaimer	3
Description	5
Project Engagement	5
Logo	5
Contract Link	6
Methodology	8
Used Code from other Frameworks/Smart Contracts (direct imports)	9
Tested Contract Files	10
Source Lines	11
Risk Level	11
Capabilities	12
Scope of Work	14
Inheritance Graph	14
Verify Claims	15
CallGraph	23
Source Units in Scope	25
Critical issues	26
High issues	26
Medium issues	26
Low issues	26
Informational issues	27
SWC Attacks	29

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	12. December 2021	<ul style="list-style-type: none">• 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



Contract Link

v1.0

TBA

Github: <https://github.com/v-Empire/vEmpire>

Testnetwork

- Game
 - <https://rinkeby.etherscan.io/address/0xa1aec39f1169a158442d3805e5b0055cdca72ab9>
- Battle
 - <https://rinkeby.etherscan.io/address/0x6d1afd11e7b0b48e9a8643080819f28630fa718a>
- xsVEMP
 - <https://rinkeby.etherscan.io/address/0x83fc8fe90FEA129FC5225Fea81FaD4f73A09692B>
- xVEMPBEP20Token
 - <https://rinkeby.etherscan.io/address/0x0F5357B6018C6f835f6bea8ab8593c99a3fD52F2#code>
- VempDao
 - <https://rinkeby.etherscan.io/address/0x1B62025D8C89E1f2F13E25Aa416f609cf99620a4>
- ProxyAdmin
 - <https://rinkeby.etherscan.io/address/0xc1838DE8F16ed5d5Da0C17e927f5d2FfeCa104dC>
- AdminUpgradeabilityProxy
 -
- Airdrop
 - <https://rinkeby.etherscan.io/address/0xB656b0a29b2D5d20D997833dfe571CEB236B9Db4>

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/v0.6/VRFConsumerBase.sol	1
@openzeppelin/contracts/GSN/Context.sol	1
@openzeppelin/contracts/access/Ownable.sol	2
@openzeppelin/contracts/math/SafeMath.sol	5
@openzeppelin/contracts/token/ERC20/ERC20Burnable.sol	1
@openzeppelin/contracts/token/ERC20/IERC20.sol	3
@openzeppelin/contracts/utils/Address.sol	1
@openzeppelin/contracts/utils/Context.sol	1

Tested Contract Files

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

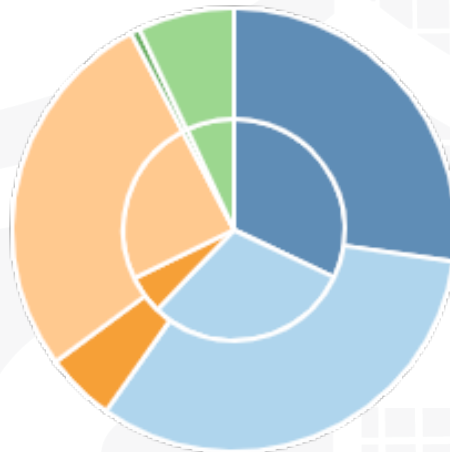
A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

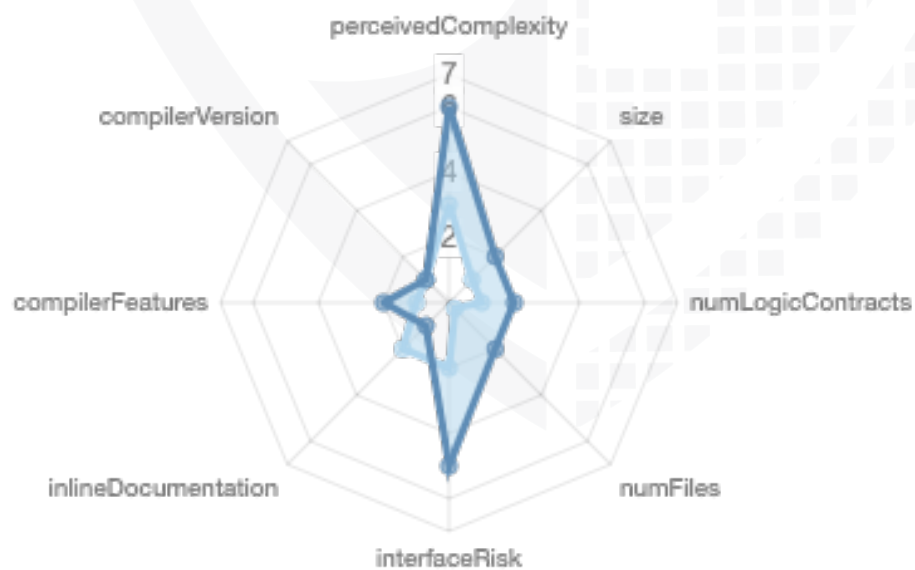
File Name	SHA-1 Hash
v3/contracts/proxy/ProxyAdmin.sol	9e85c0269351643e9b4e8b5d966d5dedd37f13dc
v3/contracts/bscdao/xVEMPBEP20Token.sol	41172247080e512bef435320e0cd7faa1cc37dbb
v3/contracts/proxy/AdminUpgradeabilityProxy.sol	778b023b0e64bfbda13b5352731cffd1e2cce180
v3/contracts/bscdao/VempDao.sol	edda6a3f523940b9701b1071ee196708d7db48d7
v3/contracts/battletoken/xsVEMP.sol	71b31a780e5ffae48c03925afe44480fea18f17d
v3/contracts/game/Game.sol	9997787b7846eda03573aaf2bf404c31dd1cc98b
v3/contracts/game/Battle.sol	c7ca12da22425cba25f53f955c978e050835ecd2
v3/contracts/game/Airdrop.sol	02acb44668d71d16bfe8ca859ecf182e347a19a8

Metrics

Source Lines v1.0



Risk Level v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	12	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	65	11

Version	External	Internal	Private	Pure	View
1.0	20	115	4	1	18

State Variables

Version	Total	Public
1.0	29	21

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	=0.6.12 ^0.6.2 ^0.6.0	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		

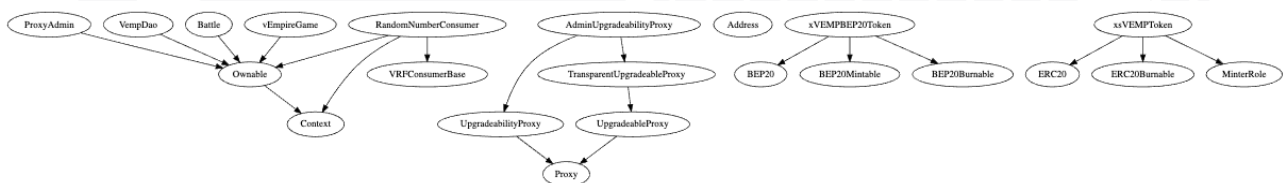
Scope of Work

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	—	—	—

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	–	–	–

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	–	–	–

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	—	—	—

Overall checkup (Smart Contract Security)

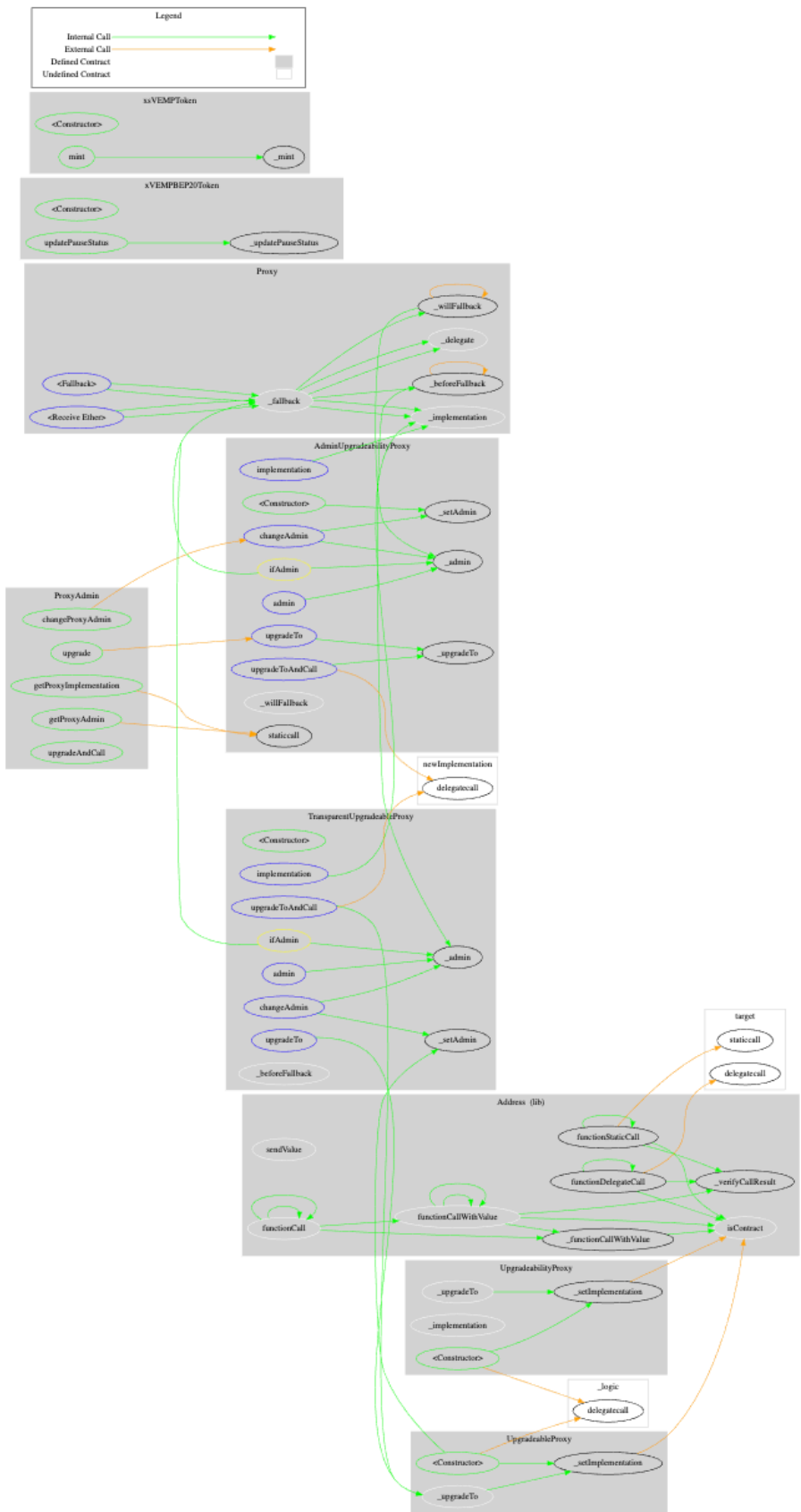
Tested	Verified
✓	✓

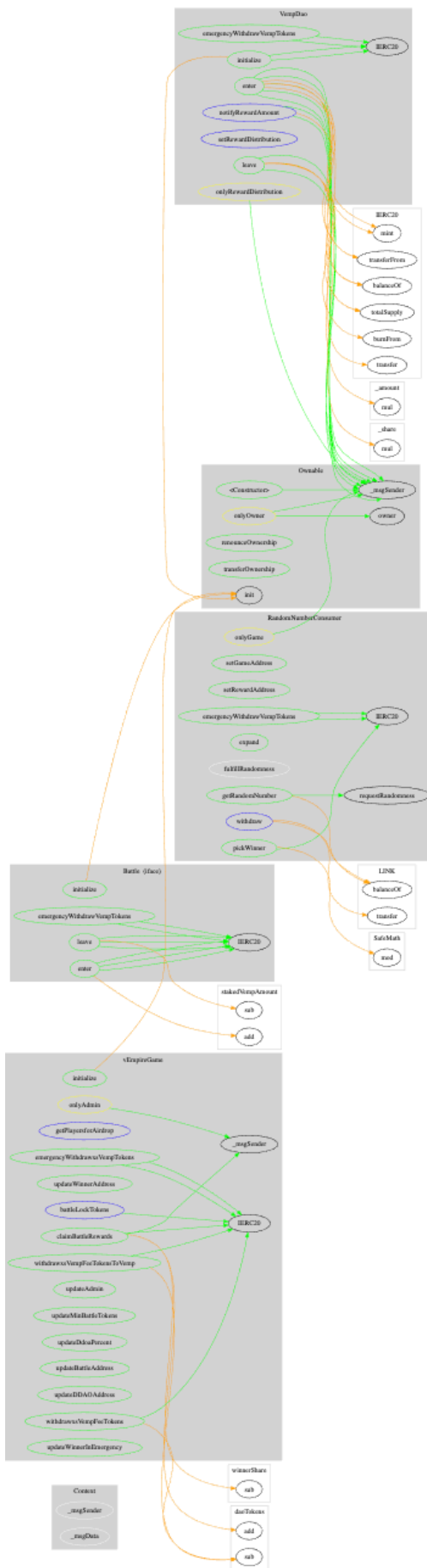
Legend

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

CallGraph



















v1.0





Source Units in Scope

v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	v3/contracts/proxy/ProxyAdmin.sol	7	————	623	595	236	342	249	
	v3/contracts/bscdao/xVEMPBEP20Token.sol	1	————	17	17	14	————	14	
	v3/contracts/proxy/AdminUpgradeabilityProxy.sol	5	————	477	455	169	259	190	
	v3/contracts/bscdao/VempDao.sol	1	————	96	87	65	10	77	
	v3/contracts/battletoken/xsVEMP.sol	1	————	31	31	17	8	14	
	v3/contracts/game/Game.sol	1	1	362	335	204	85	170	
	v3/contracts/game/Battle.sol	1	————	74	71	37	23	51	
	v3/contracts/game/Airdrop.sol	1	1	141	124	83	29	97	
	Totals	18	2	1821	1715	825	756	862	

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

Issue	File	Type	Line	Description
#1	AdminUpgradabilityProxy	A floating pragma is set	89, 233, 315, 468	The current pragma Solidity directive is „^0.6.2“.
#2	AdminUpgradabilityProxy	Missing Zero Address Validation (missing-zero-check)	253, 427	Check that the address is not zero
#3	ProxyAdmin	Missing Zero Address Validation (missing-zero-check)	607, 620	Check that the address is not zero
#4	VempDao	Missing Zero Address Validation (missing-zero-check)	67	Check that the address is not zero
#5	xVEMP BEP20Token	Local variables shadowing	9	Symbol parameter shadows BEP20.symbol Name parameter shadows BEP20.name Rename the local variables that shadow another component

#6	Airdrop	Unchecked tokens transfer	139	Use `SafeERC20`, or ensure that the transfer/transferFrom return value is checked
----	---------	---------------------------	-----	---

Informational issues

Issue	File	Type	Line	Description
#1	xVEMP BEP20Token	SPDX-License is missing	-	Consider adding a comment containing "SPDX-License-Identifier: UNLICENSED"
#2	Admin Upgradability Proxy	Functions that are not used	205	Remove unused functions
#3	BEP20	Functions that are not used	322, 467	Remove unused functions
#4	VempDAO	Naming convention	46, 56	Name the variable in such a way that it becomes understandable what this variable is
#5	Airdrop	Unused parameter	88	Remove or comment out the variable name
#6	Airdrop	CamelCase issue	93	Write <code>getPlayersForAirdrop</code> with upper case F instead of <code>getPlayersforAirdrop</code> Don't forget to change it everywhere it is used even in the interface
#7	Game	Modify error message	119	Start error message with an upper case

#8	Game	Unncessary if statement	162	<p>Following if statement is without functionality</p> <pre>if(!user.xsVempLockStatus) { ... }</pre> <p>because of line 141 and line 157 the user.xsVempLockStatus will always set to false</p>
#9	Game	Misspelling error	288	<p>Write updateDDAOPercent instead of updateDdoaPercent</p>

SWC Attacks

ID	Title	Relationships	Status
SW C-13 6	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	PASSED
SW C-13 5	Code With No Effects	CWE-1164: Irrelevant Code	NOT PASSED
SW C-13 4	Message call with hardcoded gas amount	CWE-655: Improper Initialization	PASSED
SW C-13 3	Hash Collisions With Multiple Variable Length Arguments	CWE-294: Authentication Bypass by Capture-replay	PASSED
SW C-13 2	Unexpected Ether balance	CWE-667: Improper Locking	PASSED
SW C-13 1	Presence of unused variables	CWE-1164: Irrelevant Code	PASSED
SW C-13 0	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	PASSED
SW C-12 9	Typographical Error	CWE-480: Use of Incorrect Operator	PASSED
SW C-12 8	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	PASSED

SW C-12 7	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SW C-12 5	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SW C-12 4	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SW C-12 3	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SW C-12 2	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SW C-12 1	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SW C-12 0	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	NOT 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-111	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-10 9	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	PASSED
SW C-10 8	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED
SW C-10 7	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	PASSED
SW C-10 6	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	PASSED

SW C-10 5	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	PASSED
SW C-10 4	Unchecked Call Return Value	CWE-252: Unchecked Return Value	PASSED
SW C-10 3	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	NOT PASSED
SW C-10 2	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	PASSED
SW C-10 1	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	PASSED
SW C-10 0	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