SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

Audit

Security Assessment

17. September, 2021

For

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	17. September 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 14th of September 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

Testnetwork

- MasterChefMana
    - https://kovan.etherscan.io/address/0xF60CEE1c35f2785a2C3Ba20839CD8749c2E542Bf#code
- MasterChefVemp
    - https://kovan.etherscan.io/address/0xF9dFb0f879332afA4A8c1cD2012fC65C34Ae1111#code
- MasterChefETH
    - https://kovan.etherscan.io/address/0x0C5bE007653530b5b71Bf7E99F8d77E0C60f6987#code
- MasterChefAxs
    - https://kovan.etherscan.io/address/0x454Fe4ceb6b6D03205a703595A278d27ba384d69#code
- MasterChefSAND
    - https://kovan.etherscan.io/address/0xc8e2b6f8C4b6D6e113f19e9665eaA9a4b45dFC8B#code
- MasterChefSTARL
    - https://kovan.etherscan.io/address/0x8b6547A1FB9730Df2EE783bb7C338a0533DE938B#code
- Timelock
    - https://kovan.etherscan.io/address/0x497Ae9B16ED4dae1B56Dc74cCd55aBaa2b978FaC#code
- GovernorAlpha
    - https://kovan.etherscan.io/address/0xca4869D12bc3BE75aDF19536B40772fB80c53886#code
- xVEMPToken
    - https://kovan.etherscan.io/address/0xE7AA58A6c54d5f200aCa5df0DA322E32CDBc0fE0#code
- xsVEMPToken
    - https://kovan.etherscan.io/address/0x47FB0eC09BA32997200f7705794070A342Fe6259#code

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:

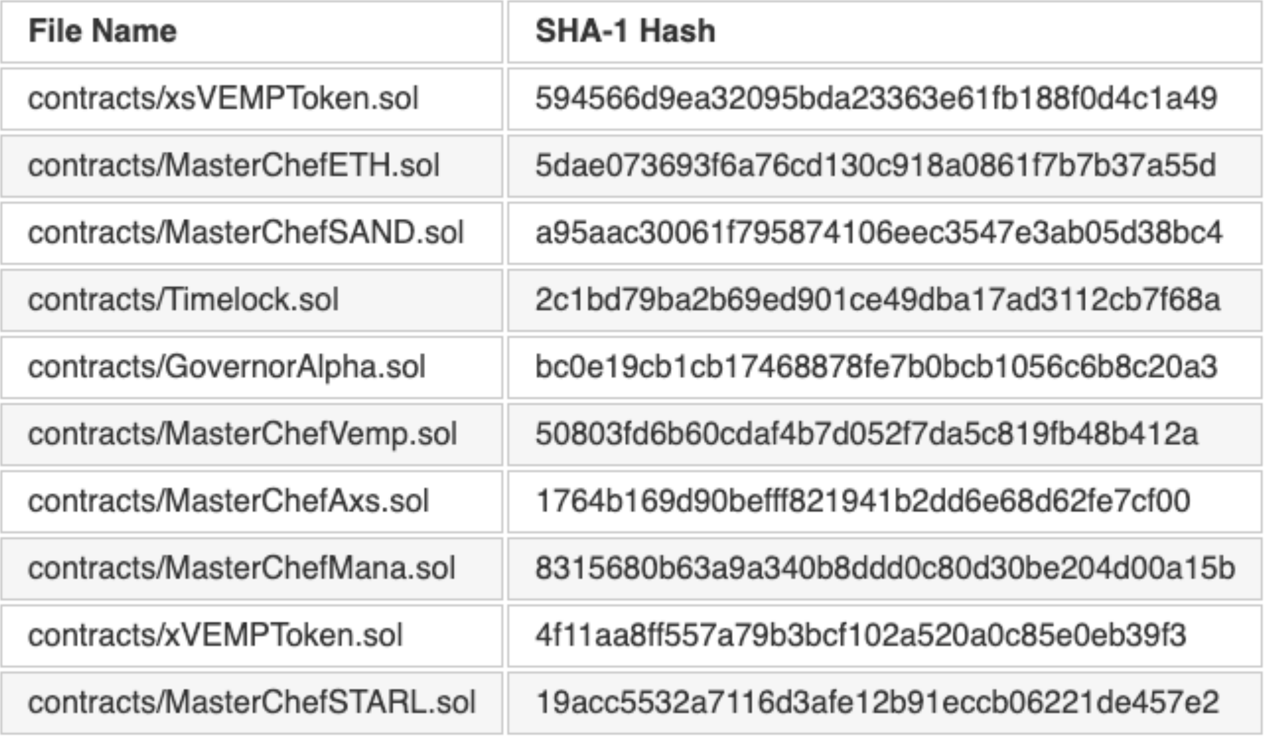
- OpenZeppelin
  - Address
  - Ownable
  - SafeMath
  - Context
  - ERC20Mintable
  - ERC20Burnable
  - IERC20
  - Roles
  - SafeERC20

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

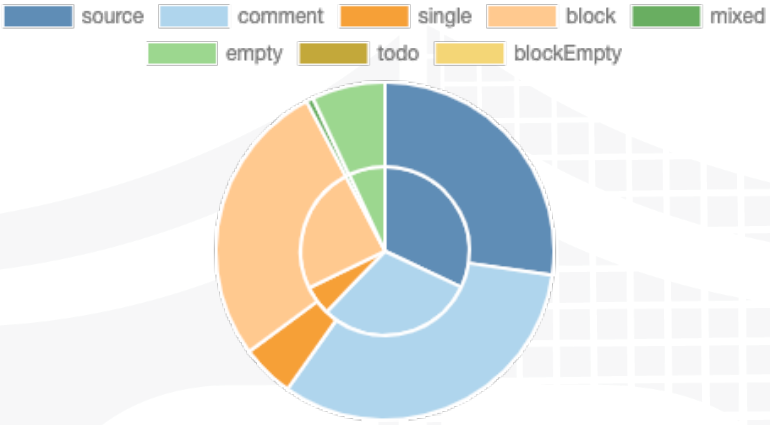


File Name	SHA-1 Hash
contracts/xsVEMPToken.sol	594566d9ea32095bda23363e61fb188f0d4c1a49
contracts/MasterChefETH.sol	5dae073693f6a76cd130c918a0861f7b7b37a55d
contracts/MasterChefSAND.sol	a95aac30061f795874106eec3547e3ab05d38bc4
contracts/Timelock.sol	2c1bd79ba2b69ed901ce49dba17ad3112cb7f68a
contracts/GovernorAlpha.sol	bc0e19cb1cb17468878fe7b0bcb1056c6b8c20a3
contracts/MasterChefVemp.sol	50803fd6b60cdaf4b7d052f7da5c819fb48b412a
contracts/MasterChefAxs.sol	1764b169d90befff821941b2dd6e68d62fe7cf00
contracts/MasterChefMana.sol	8315680b63a9a340b8ddd0c80d30be204d00a15b
contracts/xVEMPToken.sol	4f11aa8ff557a79b3bcf102a520a0c85e0eb39f3
contracts/MasterChefSTARL.sol	19acc5532a7116d3afe12b91eccb06221de457e2

Metrics

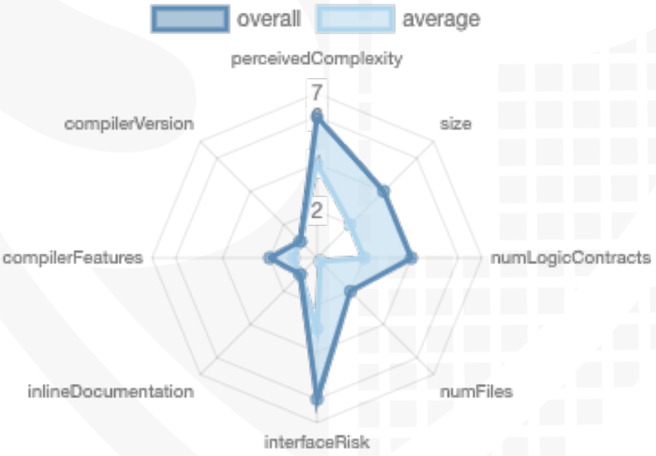
Source Lines

v1.0



Risk Level

v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	14	24	10	19

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	238	6

Version	External	Internal	Private	Pure	View
1.0	79	566	13	142	119

State Variables

Version	Total	Public
1.0	107	83

Capabilities

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

1.0	`=0.6.12`
	`^0.6.12`
		`ABIEncoderV2`	`yes`	`yes`
				(18 asm blocks)	

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

Scope of Work

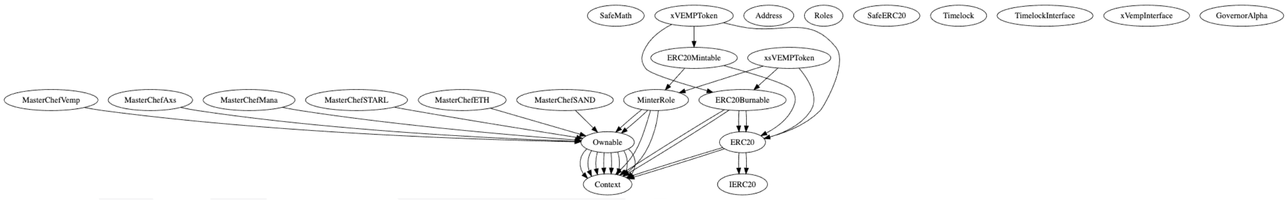
The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

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
✓	✓

MasterChefMana

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	–	–	–

MasterChefVemp

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	–	–	–

MasterChefETH

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	–	–	–

MasterChefAxs

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	–	–	–

MasterChefSAND

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	–	–	–

MasterChefSTARL

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	–	–	–

Timelock

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	–	–	–

GovernorAlpha

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	–	–	–

xVEMPToken

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	✓	✓	✓

xsVEMPToken

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	✓	✓	✓

Optional implementations

File	Function	Description	Exist	Tested	Verified
MasterChefMana	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
MasterChefVemp	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
MasterChefETH	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
MasterChefAxs	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
MasterChefSAND	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
MasterChefSTARL	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
Timelock	renounceOwnership	Owner renounce ownership for more trust	–	–	–
GovernorAlpha	renounceOwnership	Owner renounce ownership for more trust	–	–	–
xVEMPToken	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
xsVEMPToken	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗

Deployer cannot mint any new tokens

File	Name	Exist	Tested	Verified	File
MasterChefMana	Deployer cannot mint	–	–	–	Main
MasterChefVemp	Deployer cannot mint	–	–	–	Main
MasterChefETH	Deployer cannot mint	–	–	–	Main
MasterChefAxs	Deployer cannot mint	–	–	–	Main
MasterChefSAND	Deployer cannot mint	–	–	–	Main
MasterChefSTARL	Deployer cannot mint	–	–	–	Main
Timelock	Deployer cannot mint	–	–	–	Main
GovernorAlpha	Deployer cannot mint	–	–	–	Main
xVEMPToken	Deployer cannot mint	✓	✓	✗	Main
xsVEMPToken	Deployer cannot mint	✓	✓	✗	Main

Max / Total Supply:

- xVEMPToken
  - Date: 17. September 2021
  - Amount: 760.000.000.000.000.000.000
- xsVEMPToken
  - Date: 17. September 2021
  - Amount: 0

Deployer cannot burn or lock user funds

File	Name	Exist	Teste d	Verified
MasterChefMana	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
MasterChefVemp	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
MasterChefETH	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
MasterChefAxs	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
MasterChefSAND	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
MasterChefSTARL	Deployer cannot lock	–	–	–
	Deployer cannot burn	–	–	–
Timelock	Deployer cannot lock	✓	✓	✓
	Deployer cannot burn	–	–	–
GovernorAlpha	Deployer cannot lock	–	–	–

	Deployer cannot burn	–	–	–
xVEMPToken	Deployer cannot lock	–	–	–
	Deployer cannot burn	✓	✓	✗
xsVEMPToken	Deployer cannot lock	–	–	–
	Deployer cannot burn	✓	✓	✗

Deployer cannot pause the contract

File	Name	Exist	Tested	Verified
MasterChefMana	Deployer cannot pause	✓	✓	✓
MasterChefVemp	Deployer cannot pause	✓	✓	✓
MasterChefETH	Deployer cannot pause	✓	✓	✓
MasterChefAxs	Deployer cannot pause	✓	✓	✓
MasterChefSAND	Deployer cannot pause	✓	✓	✓
MasterChefSTARL	Deployer cannot pause	✓	✓	✓
Timelock	Deployer cannot pause	✓	✓	✓
GovernorAlpha	Deployer cannot pause	✓	✓	✓
xVEMPToken	Deployer cannot pause	✓	✓	✓
xsVEMPToken	Deployer cannot pause	✓	✓	✓

Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

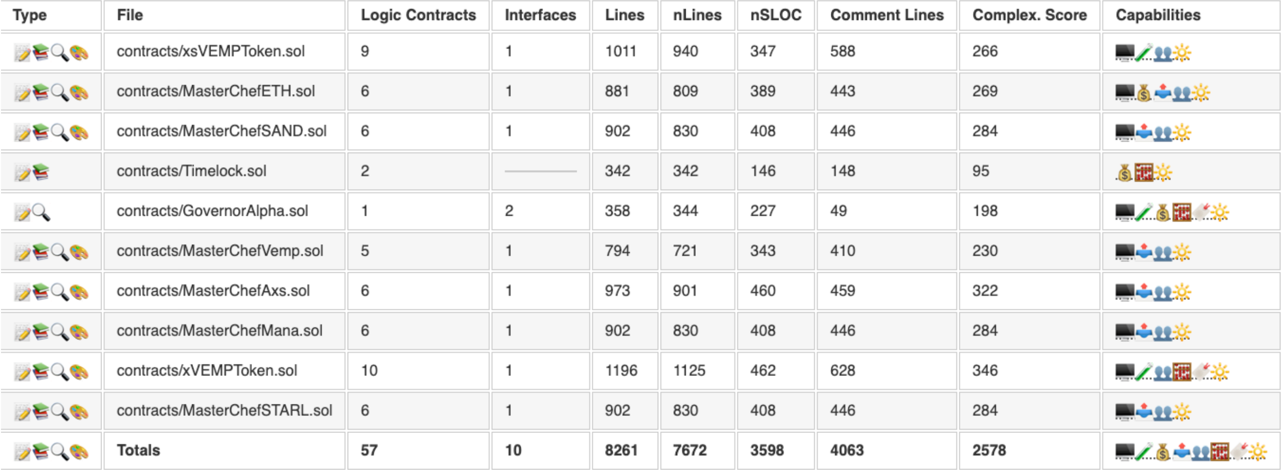
Legend

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

CallGraph

Source Units in Scope

v1.0



Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
📝📚🔍🎨	contracts/xsVEMPToken.sol	9	1	1011	940	347	588	266	🖥🧪👥☀️
📝📚🔍🎨	contracts/MasterChefETH.sol	6	1	881	809	389	443	269	🖥💰📤👥☀️
📝📚🔍🎨	contracts/MasterChefSAND.sol	6	1	902	830	408	446	284	🖥📤👥☀️
📝📚	contracts/Timelock.sol	2	----	342	342	146	148	95	💰🧮☀️
📝🔍	contracts/GovernorAlpha.sol	1	2	358	344	227	49	198	🖥🧪💰🧮🔖☀️
📝📚🔍🎨	contracts/MasterChefVemp.sol	5	1	794	721	343	410	230	🖥📤👥☀️
📝📚🔍🎨	contracts/MasterChefAxs.sol	6	1	973	901	460	459	322	🖥📤👥☀️
📝📚🔍🎨	contracts/MasterChefMana.sol	6	1	902	830	408	446	284	🖥📤👥☀️
📝📚🔍🎨	contracts/xVEMPToken.sol	10	1	1196	1125	462	628	346	🖥🧪👥🧮🔖☀️
📝📚🔍🎨	contracts/MasterChefSTARL.sol	6	1	902	830	408	446	284	🖥📤👥☀️
📝📚🔍🎨	**Totals**	**57**	**10**	**8261**	**7672**	**3598**	**4063**	**2578**	🖥🧪💰📤👥🧮🔖☀️

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	Main	Contract doesn’t import npm packages from source (like OpenZeppelin etc.)	-	We recommend to import all packages from npm directly without flatten the contract. Functions could be modified or can be susceptible to vulnerabilities
#2	GovernorAlpha.sol	Missing Zero Address Validation (missing-zero-check)	152	Check that the address is not zero
#3	GovernorAlpha.sol	State variable visibility is not set	31, 32	It is best practice to set the visibility of state variables explicitly
#4	MasterChefAxs.sol	A floating pragma is set	7	The current pragma Solidity directive is „“^0.6.12““.
#5	Timelock.sol	Missing Zero Address Validation (missing-zero-check)	311	Check that the address is not zero

Informational issues

Issue	File	Type	Line	Description

#1	xsVEMPToken.sol	Functions that are not used (dead-code)	853-855	Remove unused functions

Audit Comments

17. September 2021:

- Testnetwork:
    - There is still an owner (Owner still has not renounced ownership)

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	**NOT 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

Solid Proofed

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY