



**SOLIDProof**  
*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC**

MADE IN GERMANY

# Audit

**Security Assessment**  
**27. August, 2021**

**For**



Disclaimer	3
Description	5
Project Engagement	5
Logo	5
Contract Link	5
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	24
Critical issues	25
High issues	25
Medium issues	25
Low issues	25
Informational issues	26
Commented Code exist	26
Audit Comments	27
SWC Attacks	28

# 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	27. August 2021	<ul style="list-style-type: none"><li>• Layout project</li><li>• Automated- /Manual-Security Testing</li><li>• Summary</li></ul>

## **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 25th of August 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

- MockToken
  - <https://kovan.etherscan.io/address/0x3d8DAa0C319623Df0f80aaA832ecf539096E9dd0#code>
- xVEMPToken
  - <https://kovan.etherscan.io/address/0x240D648beE8a4db6fC454883e71a45bbC2f6518C#code>
- TestMANA
  - <https://kovan.etherscan.io/address/0x3e23cf68E78291646B9297AF451896A508DC7dE9#code>
- MasterChefMana
  - <https://kovan.etherscan.io/address/0x88744DDDAe7512F768Fdf59266f350ab7A81af8D#code>
- MasterChefVemp
  - <https://kovan.etherscan.io/address/0x3F9C56925365574a17B84E87CC638153202A1f3E#code>
- Timelock
  - <https://kovan.etherscan.io/address/0xE0442cc439348f399776fcc10D79B058866f5f65#code>
- GovernorAlpha
  - <https://kovan.etherscan.io/address/0x3A08B8F1A54Ba9155D8F5681dc1eec3da39ecA5b#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)
<b>Critical</b>	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.
<b>High</b>	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.
<b>Medium</b>	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.
<b>Low</b>	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.
<b>Informational</b>	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.

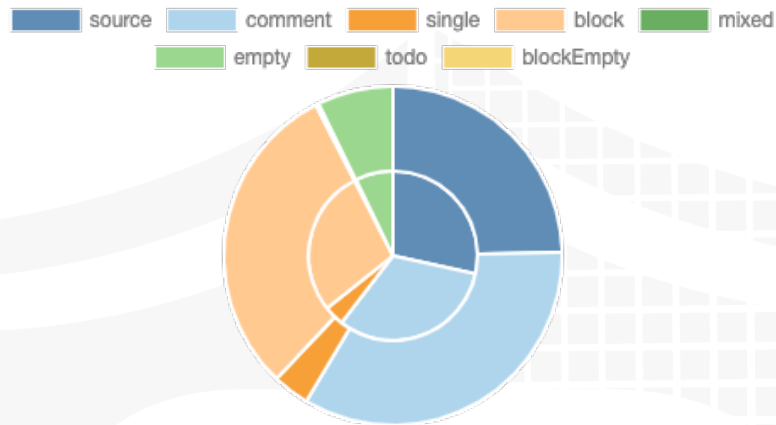
*A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.*

### v1.0

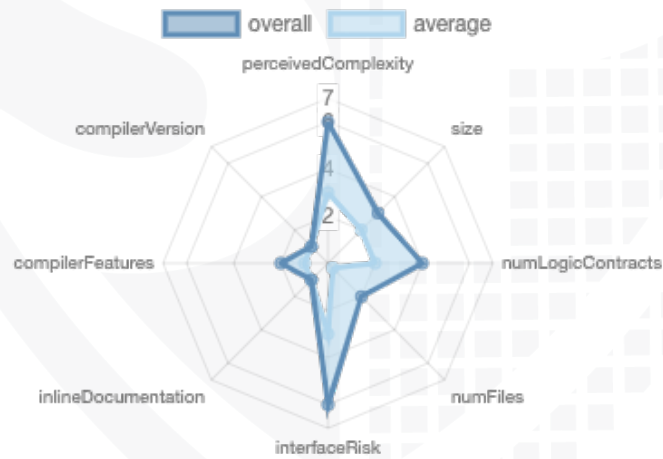
File Name	SHA-1 Hash
contracts/xVEMP.sol	c25587a7c918dca44d3428091f5bbd7d24a18df3
contracts/Timelock.sol	44b1ac2112fd59eec412c6ec8d8e52aeb5a41c37
contracts/GovernorAlpha.sol	c0c7f85f7274a85c3873a9a4b89b26d4385fa411
contracts/MasterChefVemp.sol	dd2dcd020883f3ea5c42b1d67593d2d3b22508f6
contracts/TestMana.sol	1df1a9814fc86a9a9b799a15ac5049a806cce6c
contracts/MockToken.sol	1df1a9814fc86a9a9b799a15ac5049a806cce6c
contracts/MasterChefMana.sol	99ab930df1c90050803fcc670607fcff339c28b6

# Metrics

## Source Lines v1.0



## Risk Level v1.0



# Capabilities

## Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	11	11	7	12

## 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	154	4

Version	External	Internal	Private	Pure	View
1.0	48	315	4	89	72

## State Variables

Version	Total	Public
1.0	62	40

## Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	<code>^0.6.12</code>	ABIEncoderV2	yes	yes (8 asm blocks)	

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

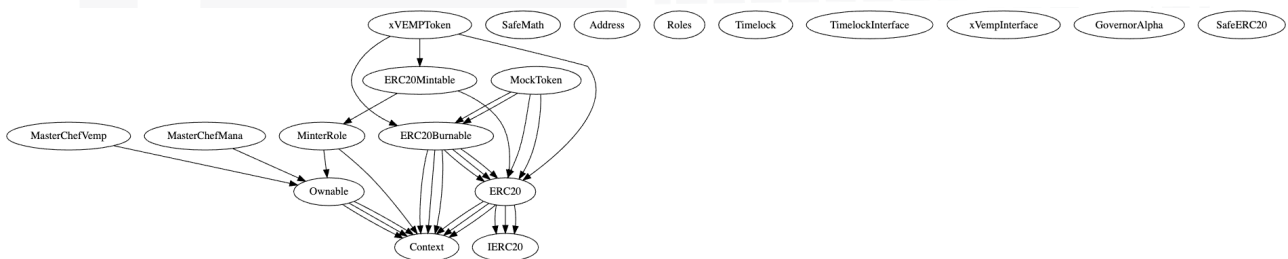
## Scope of Work

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
✓	✓

#### MockToken

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	✓	✓	✓
-----------	--	---	---	---

## TestMANA

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	✓	✓	✓

## 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	—	—	—

## 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	–	–	–

## Optional implementations

File	Function	Description	Exist	Tested	Verified
MockToken	renounceOwnership	Owner renounce ownership for more trust	–	–	–
xVEMPToken	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
TestMANA	renounceOwnership	Owner renounce ownership for more trust	–	–	–
MasterChefMana	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
MasterChefVemp	renounceOwnership	Owner renounce ownership for more trust	✓	✓	✗
Timelock	renounceOwnership	Owner renounce ownership for more trust	–	–	–
GovernorAlpha	renounceOwnership	Owner renounce ownership for more trust	–	–	–

## Deployer cannot mint any new tokens

File	Name	Exist	Tested	Verified	File
MockToken	Deployer cannot mint	✓	✓	✗	Main
xVEMPToken	Deployer cannot mint	✓	✓	✗	Main
TestMANA	Deployer cannot mint	✓	✓	✗	Main
MasterChefMan a	Deployer cannot mint	—	—	—	Main
MasterChefVem p	Deployer cannot mint	—	—	—	Main
Timelock	Deployer cannot mint	—	—	—	Main
GovernorAlpha	Deployer cannot mint	—	—	—	Main

### Max / Total Supply:

- MockToken
  - Date: 27. August 2021, 2:18pm
  - Amount: 2307970000000000000000
- xVEMPToken
  - Date: 27. August 2021, 2:18pm
  - Amount: 1567100100000000000000
- TestMANA
  - Date: 27. August 2021, 2:18pm
  - Amount: 4200000000000000000000

## Deployer cannot burn or lock user funds

File	Name	Exist	Tested	Verified
MockToken	Deployer cannot lock	-	-	-
	Deployer cannot burn	✓	✓	✗
xVEMPToken	Deployer cannot lock	-	-	-
	Deployer cannot burn	✓	✓	✗
TestMANA	Deployer cannot lock	-	-	-
	Deployer cannot burn	✓	✓	✗
MasterChefMa na	Deployer cannot lock	-	-	-
	Deployer cannot burn	-	-	-
MasterChefVe mp	Deployer cannot lock	-	-	-
	Deployer cannot burn	-	-	-
Timelock	Deployer cannot lock	✓	✓	✓
	Deployer cannot burn	-	-	-
GovernorAlpha	Deployer cannot lock	-	-	-
	Deployer cannot burn	-	-	-

## Deployer cannot pause the contract

File	Name	Exist	Tested	Verified
MockToken	Deployer cannot pause	✓	✓	✓
xVEMPToken	Deployer cannot pause	✓	✓	✓
TestMANA	Deployer cannot pause	✓	✓	✓
MasterChefM ana	Deployer cannot pause	✓	✓	✓
MasterChefVe mp	Deployer cannot pause	✓	✓	✓
Timelock	Deployer cannot pause	✓	✓	✓
GovernorAlph a	Deployer cannot pause	✓	✓	✓

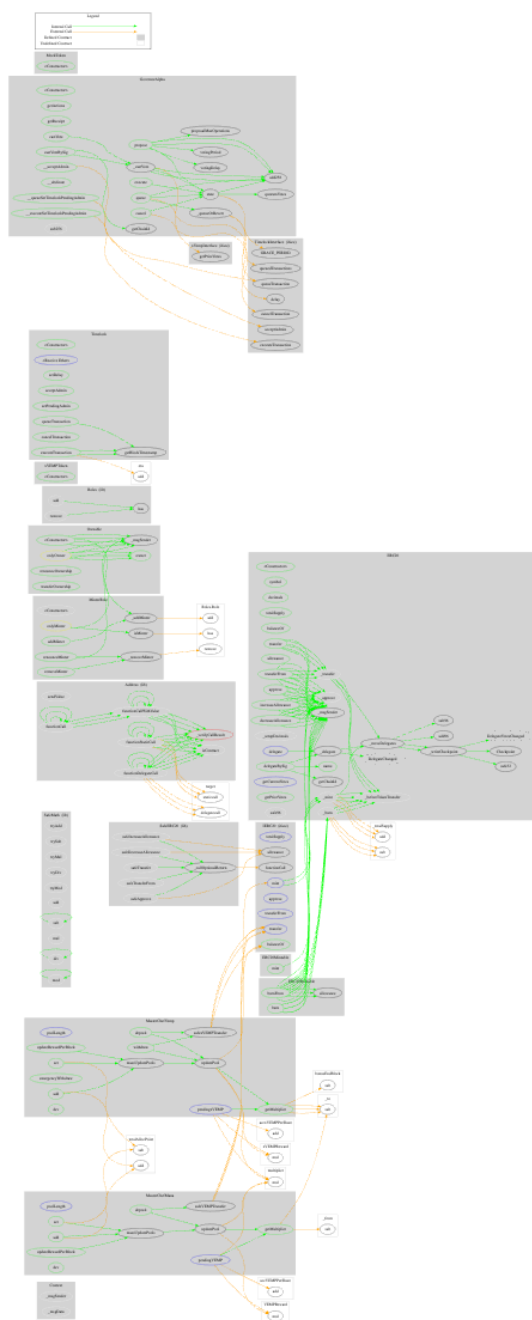
# Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

## Legend

Attribute	Symbol
Verified / 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/xVEMP.sol	10	1	1196	1125	462	628	346	
	contracts/Timelock.sol	2	—	286	286	123	114	85	
	contracts/GovernorAlpha.sol	1	2	340	327	216	45	178	
	contracts/MasterChefVemp.sol	5	1	790	717	343	407	230	
	contracts/TestMana.sol	5	1	667	616	193	419	139	
	contracts/MockToken.sol	5	1	667	616	193	419	139	
	contracts/MasterChefMana.sol	6	1	825	753	345	436	226	
	<b>Totals</b>	<b>34</b>	<b>7</b>	<b>4771</b>	<b>4440</b>	<b>1875</b>	<b>2468</b>	<b>1343</b>	

## 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	xVEMP.sol	A floating pragma is set	7	The current pragma Solidity directive is „^0.6.12“.
#3	Timelock.sol	A floating pragma is set	4	The current pragma Solidity directive is „^0.6.12“.
#4	TestMana.sol	A floating pragma is set	7	The current pragma Solidity directive is „^0.6.12“.
#5	MockToken.sol	A floating pragma is set	7	The current pragma Solidity directive is „^0.6.12“.
#6	MasterChefVemp.sol	A floating pragma is set	7	The current pragma Solidity directive is „^0.6.12“.
#7	MasterChefMana.sol	A floating pragma is set	7	The current pragma Solidity directive is „^0.6.12“.
#8	GovernorAlpha.sol	A floating pragma is set	7	The current pragma Solidity directive is „^0.6.12“.
#9	GovernorAlpha.sol	Missing Zero Address Validation (missing-zero-check)	152	Check that the address is not zero

#10	Timelock.sol	Missing Zero Address Validation (missing-zero-check)	193, 255, 222	Check that the address is not zero
-----	--------------	--	---------------	------------------------------------

## Informational issues

Issue	File	Type	Line	Description
#1	MasterChefMania.sol	Functions that are not used (dead-code)	330-333	Remove unused functions

## Commented Code exist

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

File	Line	Comment
Timelock.sol	126	// assert(a == b * c + a % b);

## Recommendation

Remove the commented code, or address them properly.

## Audit Comments

### 27. August 2021:

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



## SWC Attacks

ID	Title	Relationships	Status
<a href="#">SW C-13 6</a>	Unencrypted Private Data On-Chain	<a href="#">CWE-767: Access to Critical Private Variable via Public Method</a>	PASSED
<a href="#">SW C-13 5</a>	Code With No Effects	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-13 4</a>	Message call with hardcoded gas amount	<a href="#">CWE-655: Improper Initialization</a>	PASSED
<a href="#">SW C-13 3</a>	Hash Collisions With Multiple Variable Length Arguments	<a href="#">CWE-294: Authentication Bypass by Capture-replay</a>	PASSED
<a href="#">SW C-13 2</a>	Unexpected Ether balance	<a href="#">CWE-667: Improper Locking</a>	PASSED
<a href="#">SW C-13 1</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-13 0</a>	Right-To-Left-Override control character (U+202E)	<a href="#">CWE-451: User Interface (UI) Misrepresentation of Critical Information</a>	PASSED
<a href="#">SW C-12 9</a>	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	PASSED
<a href="#">SW C-12 8</a>	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	PASSED

<a href="#">SW C-12 7</a>	Arbitrary Jump with Function Type Variable	<a href="#">CWE-695: Use of Low-Level Functionality</a>	<b>PASSED</b>
<a href="#">SW C-12 5</a>	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	<b>PASSED</b>
<a href="#">SW C-12 4</a>	Write to Arbitrary Storage Location	<a href="#">CWE-123: Write-what-where Condition</a>	<b>PASSED</b>
<a href="#">SW C-12 3</a>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	<b>PASSED</b>
<a href="#">SW C-12 2</a>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	<b>PASSED</b>
<a href="#">SW C-12 1</a>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>
<a href="#">SW C-12 0</a>	Weak Sources of Randomness from Chain Attributes	<a href="#">CWE-330: Use of Insufficiently Random Values</a>	<b>PASSED</b>
<a href="#">SW C-11 9</a>	Shadowing State Variables	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-11 8</a>	Incorrect Constructor Name	<a href="#">CWE-665: Improper Initialization</a>	<b>PASSED</b>
<a href="#">SW C-11 7</a>	Signature Malleability	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>

<a href="#">SW C-11 6</a>	Timestamp Dependence	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-11 5</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 4</a>	Transaction Order Dependence	<a href="#">CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')</a>	<b>PASSED</b>
<a href="#">SW C-11 3</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	<b>PASSED</b>
<a href="#">SW C-11 2</a>	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-111</a>	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 0</a>	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	<b>PASSED</b>
<a href="#">SW C-10 9</a>	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	<b>PASSED</b>
<a href="#">SW C-10 8</a>	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-10 7</a>	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	<b>PASSED</b>
<a href="#">SW C-10 6</a>	Unprotected SELFDESTRUCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>

<a href="#">SW C-10 5</a>	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	PASSED
<a href="#">SW C-10 4</a>	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	PASSED
<a href="#">SW C-10 3</a>	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	NOT PASSED
<a href="#">SW C-10 2</a>	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	PASSED
<a href="#">SW C-10 1</a>	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	PASSED
<a href="#">SW C-10 0</a>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	PASSED

The logo features the words "Solid Proofed" in a white, elegant script font. The text is superimposed on a dark blue background that contains a faint, stylized shield emblem. The shield has a grid-like pattern and is partially obscured by the text.

Solid  
Proofed

**Blockchain Security | Smart Contract Audits | KYC**

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

MADE IN GERMANY