Cyberscope

Audit Report

PayMe Vesting

December 2022

Github https://github.com/payMeQuiz/payMe-Project

Commit 03d379a3e1cd42e94bbb4c6fb0c02f31ea9ac3b5

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	PaymeTokenVesting
Repository	https://github.com/PayMe-Platforms-Token/ICO/tree/master/contracts
Commit	03d379a3e1cd42e94bbb4c6fb0c02f31ea9ac3b5
Testing Deploy	https://testnet.bscscan.com/address/0x52f41dc9172402f03f9c4478b5cfedaa7de764e7

Audit Updates

Initial Audit	18 Oct 2022 https://github.com/cyberscope-io/audits/blob/main/payme/v1/paymeTokenVesting.pdf
Corrected Phase 2	19 Oct 2022 https://github.com/cyberscope-io/audits/blob/main/payme/v2/paymeTokenVesting.pdf
Corrected Phase 3	09 Nov 2022 https://github.com/cyberscope-io/audits/blob/main/payme/v3/paymeTokenVesting.pdf
Corrected Phase 4	10 Feb 2023

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/security/ReentrancyGuard.sol	aa73590d5265031c5bb64b5c0e7f84c44cf5f8539e6d8606b763adac784e8b2e
@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol	3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	fa36a21bd954262006d806b988e4495562e7b50420775e2aa0deecb596fd1902
@openzeppelin/contracts/utils/Address.sol	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/math/Math.sol	929523c09910460ad708c75878d89b9fbed12b65cb5d8b670200c793131072f4
@openzeppelin/contracts/utils/math/SafeMath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a
contracts/PaymeTokenVesting.sol	b29f2717dd9ef528fe0758cbf340b703942f44bd8faa92cd40d12ff21d5ebc3e

Introductions

The PaymeTokenVesting contract implements a vesting contract as an upgradable proxy. The contract is responsible for creating and configuring vesting schedules for a beneficiary.
Each beneficiary can have multiple vesting schedules. In addition, the contract monitors the vesting schedules by keeping track of the beneficiaries and how many times its beneficiary has vested.

Roles

The contract has an owner role and a beneficiary role. The beneficiary is any user that vests on the contract. The owner has the authority to withdraw a specific amount from the contract if possible. Additionally, the owner and any user that is beneficiary have the authority:

1. Revoke all the vested amount if the vesting period is elapsed or the proportional amount in relation to the vested period.
2. Release tokens for TGE If the TGE opening time has elapsed.
3. Release a specific amount of vested tokens if it is possible.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	L11	Unnecessary Boolean equality	Unresolved
●	L16	Validate Variable Setters	Unresolved
●	L18	Multiple Pragma Directives	Unresolved

L11 - Unnecessary Boolean equality

Criticality	Minor / Informative
Location	contracts/PaymeTokenVesting.sol#L459
Status	Unresolved

Description

Boolean equality is unnecessary when comparing two boolean values. This is because a boolean value is either true or false, and there is no need to compare two values that are already known to be either true or false.

it's important to be aware of the types of variables and expressions that are being used in the contract's code, as this can affect the contract's behavior and performance. The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
(currentTime < vestingSchedule.cliff) || vestingSchedule.revoked == true
```

Recommendation

Using the boolean value itself is clearer and more concise, and it is generally considered good practice to avoid unnecessary boolean equalities in Solidity code.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	contracts/PaymeTokenVesting.sol#L183
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
crowdsalesAddress = icrowdsalesAddress
```

Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

L18 - Multiple Pragma Directives

Criticality	Minor / Informative
Location	contracts/PaymeTokenVesting.sol#L3
Status	Unresolved

Description

If the contract includes multiple conflicting pragma directives, it may produce unexpected errors. To avoid this, it's important to include the correct pragma directive at the top of the contract and to ensure that it is the only pragma directive included in the contract.

```
pragma solidity ^0.8.0;
```

Recommendation

It is important to include only one pragma directive at the top of the contract and to ensure that it accurately reflects the version of Solidity that the contract is written in.

By including all required compiler options and flags in a single pragma directive, the potential conflicts could be avoided and ensure that the contract can be compiled correctly.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ReentrancyGuard	Implementation			
		Public	✓	-
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-

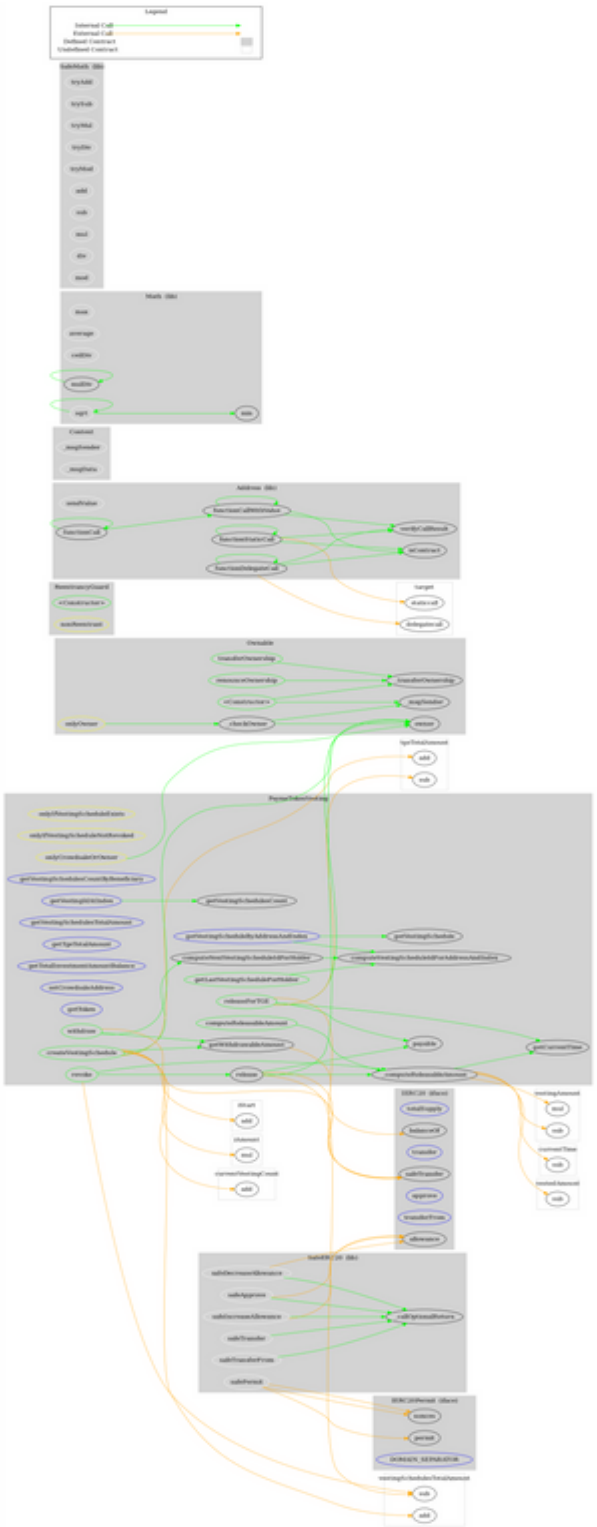
	transferFrom	External	✓	-
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	safePermit	Internal	✓	
	_callOptionalReturn	Private	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		

Math	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
	mulDiv	Internal		
	mulDiv	Internal		
	sqrt	Internal		
	sqrt	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
PaymeTokenVesting	Implementation	Ownable, ReentrancyGuard		
		Public	✓	-
	getVestingSchedulesCountByBenefic	External		-

	iary			
	getVestingIdAtIndex	External		-
	getVestingScheduleByAddressAndIndex	External		-
	getVestingSchedulesTotalAmount	External		-
	getTgeTotalAmount	External		-
	getTotalInvestmentAmountBalance	External		-
	setCrowdsaleAddress	External	✓	-
	getToken	External		-
	createVestingSchedule	Public	✓	onlyCrowdsaleOrOwner
	revoke	Public	✓	onlyOwner onlyIfVestingScheduleNotRevoked
	withdraw	Public	✓	nonReentrant onlyOwner
	releaseForTGE	Public	✓	nonReentrant
	release	Public	✓	nonReentrant onlyIfVestingScheduleNotRevoked
	getVestingSchedulesCount	Public		-
	computeReleasableAmount	Public		onlyIfVestingScheduleNotRevoked
	getVestingSchedule	Public		-
	getWithdrawableAmount	Public		-
	computeNextVestingScheduleIdForHolder	Public		-
	getLastVestingScheduleForHolder	Public		-
	computeVestingScheduleIdForAddressAndIndex	Public		-
	_computeReleasableAmount	Internal		
	getCurrentTime	Public		-

Inheritance Graph

Contract Flow



Summary

The PaymeTokenVesting contract is responsible for generating vesting schedules. This audit investigates security issues and mentions business logic concerns and potential improvements.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment 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 Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io