Cyberscope

Audit Report

PayMe Vesting

December 2022

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

Commit 0dc29331c643bfaa1e71a51b8605ae6f6f8819b5

Audited by © cyberscope

Table of Contents

Table of Contents 1
Contract Review 2
Audit Updates 2
Source Files 3
Introductions 5
Roles 5
Contract Diagnostics 6
VTI - Vesting Token Issues 7
Description 7
Recommendation 8
L05 - Unused State Variable 9
Description 9
Recommendation 9
Contract Functions 10
Contract Flow 15
Domain Info 16
Summary 17
Disclaimer 18
About Cyberscope 19

Contract Review

Contract Name	payMETokenVesting
Compiler Version	v0.8.9+commit.e5eed63a
Github	https://github.com/payMeQuiz/payMe-Project
Commit	0dc29331c643bfaa1e71a51b8605ae6f6f8819b5
Testing Deploy	https://testnet.bscscan.com/token/0xD97c59Db0A0298d44De64f14DFcf2dF72d96008B
Domain	https://payme.games

Audit Updates

Initial Audit	17th October 2022 https://github.com/cyberscope-io/audits/blob/main/payme/v1/paymeTokenVesting.pdf
Corrected Phase 1	9th November 2022 https://github.com/cyberscope-io/audits/blob/main/payme/v2/paymeTokenVesting.pdf
Corrected Phase 2	8th December 2022

Source Files

Filename	SHA256
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	da66c17044345dc892d85bd7ddc9745d25df0b3dacfba8f84eb87c60d6e40fe3
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	74def996fd6faf32f13ab9cacfc71d57400177de340fe5d5d7c6e805dfbab3bd
@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol	fcfc8be28dd0e725a6c61648b3c7a422f0e668ad2eb83c39c3c07f590846523a
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol	b97515a88e75c313eacf0a27c9439ef371d86d4c2730d3b13076640942f813df
@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol	4e09a7479aa3e7c313f8fc141c4c8fc04e0abfeb8754615ef7d78ec94c298b07
@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol	45b47dd617d02875a7e6c896d1842ff9d8362ab15b8180645f3f4b180d4f028f

@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol	1d7d481b79fd54d957c9a0696f6227f7799fec01d8ba41f5c130a7cc6b4eddc9
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	5fb301961e45cb482fe4e05646d2f529aa449fe0e90c6671475d6a32356fa2d4
@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol	158a0316fa289fad12c2ca764449e43e6724fb79c58fc438508d116f9af46b39
@openzeppelin/contracts-upgradeable/utils/math/SafeMathUpgradeable.sol	4039686a509394aed475619c4e0b3a2df1df34fe59e90b9add8669de371eb731
contracts/ico/PaymeTokenVesting.sol	89c8bf653bb3f61a0b95fad57f366f188bd72131f58878afb3482732008a1b22

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
Medium	VTI	Vesting Token Issues	Unresolved
Minor / Informative	L05	Unused State Variable	Unresolved

VTI - Vesting Token Issues

Criticality	medium
Location	contract.sol#L273
Status	Unresolved

Description

The contract can cause the `vestingSchedule.released` variable to aggregate more than the `vestingSchedule.amountTotal`. As a result, the subtraction between the amountTotal and the released will revert. This may happen in multiple ways since the `_computeReleasableAmount()` may produce a greater number than the `amountTotal`.

For instance, let's assume that a user has Vested 100 tokens and the TGE percent is 30% and the TGE opens at 51% of the vest duration.

1. The user releases the vested token at 50% of the vested time. So, the user will release 50 tokens.

```
//vestingSchedule.released = 50
```

2. The user releases the TGE amount at 51% of the vested time. The TGE amount is 30 tokens. So, the released tokens will aggregate to 80.

```
 uint256 TGEReleaseAmount =
vestingSchedule.amountTotal.mul(tgePercent).div(100); // 20

//vestingSchedule.released = 80
```

3. The user releases the vested token at 52% of the vested time. At that point, the released tokens are greater than the releasable amount. Hence the method call will revert.

```
//vestingSchedule.released = 80

function _computeReleasableAmount(VestingSchedule memory vestingSchedule)
internal view returns(uint256){
            uint256 tgeReleasableAmount = 0;
            uint256 tgeAmount =
vestingSchedule.amountTotal.mul(tgePercent).div(100);// 30
            uint256 vestingAmount = vestingSchedule.amountTotal;//100

            if(currentTime > tgeOpeningTime &&
tgeTokenParticipants[vestingScheduleId] == 0 && vestingSchedule.releaseAtTGE
&& !vestingSchedule.revoked){//true
               tgeReleasableAmount = tgeAmount;
               vestingAmount=vestingSchedule.amountTotal.sub(tgeAmount);//70
            }

            if ((currentTime < vestingSchedule.cliff) ||
vestingSchedule.revoked) {//false
                return tgeReleasableAmount;
            } else if (currentTime >=
vestingSchedule.start.add(vestingSchedule.duration)) {//false
               //time has elapsed -> release all
               return
vestingAmount.add(tgeReleasableAmount).sub(vestingSchedule.released);
            } else {
                uint256 vestedAmount =
vestingAmount.mul(timeFromStart).div(vestingSchedule.duration);// 70*52%
                vestedAmount =
vestedAmount.add(tgeReleasableAmount).sub(vestingSchedule.released);
                return vestedAmount;
                //vestedAmount = (70*52%).add(30).sub(80);// reverts
                return vestedAmount;
            }
```

Recommendation

The team is advised to check if the implementation follows the expected business logic carefully.

L05 - Unused State Variable

Criticality	minor / informative
Location	@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#L80
Status	Unresolved

Description

There are segments that contain unused state variables.

```
__gap
```

Recommendation

Remove unused state variables.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	__Ownable_init	Internal	✓	onlyInitializing
	__Ownable_init_unchained	Internal	✓	onlyInitializing
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Initializable	Implementation			
	_disableInitializers	Internal	✓	
	_getInitializedVersion	Internal		
	_isInitializing	Internal		
ReentrancyGuardUpgradeable	Implementation	Initializable		
	__ReentrancyGuard_init	Internal	✓	onlyInitializing
	__ReentrancyGuard_init_unchained	Internal	✓	onlyInitializing
	_nonReentrantBefore	Private	✓	
	_nonReentrantAfter	Private	✓	
IERC20PermitUpgradeable	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-

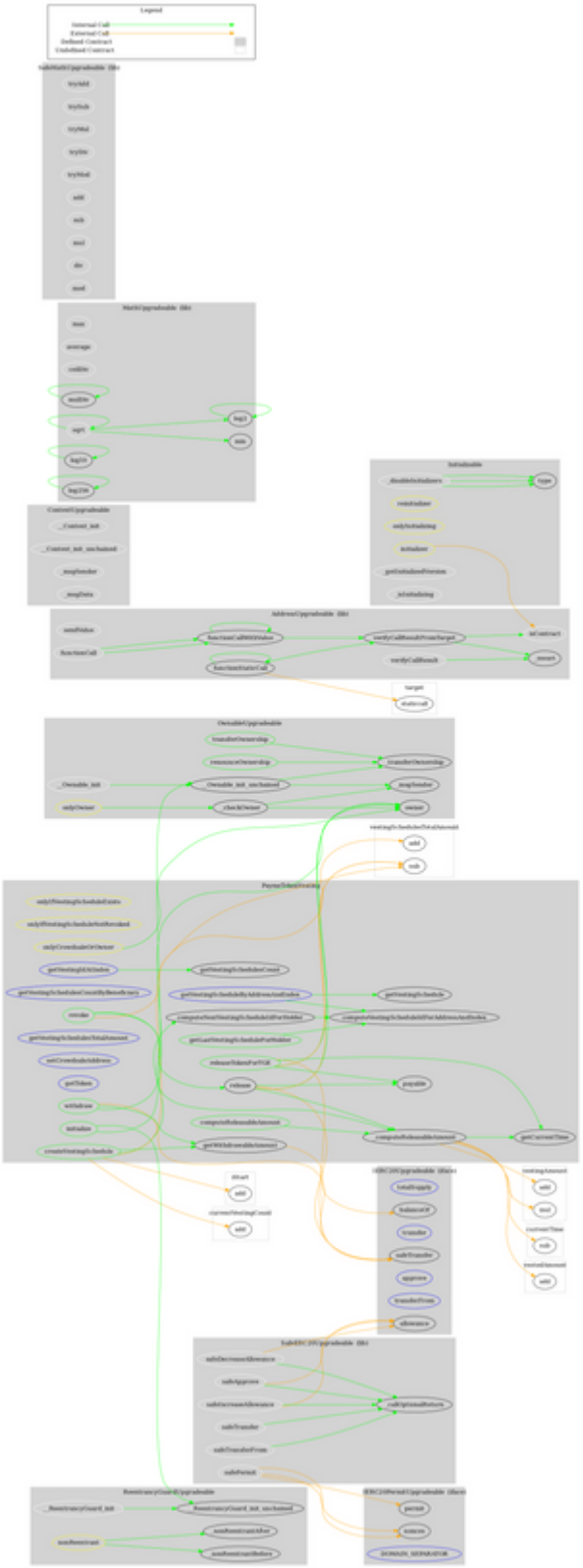
IERC20Upgradeable	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeERC20Upgradeable	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	safePermit	Internal	✓	
	_callOptionalReturn	Private	✓	
AddressUpgradeable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResultFromTarget	Internal		
	verifyCallResult	Internal		
	_revert	Private		

ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	onlyInitializing
	__Context_init_unchained	Internal	✓	onlyInitializing
	_msgSender	Internal		
	_msgData	Internal		
MathUpgradeable	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
	mulDiv	Internal		
	mulDiv	Internal		
	sqrt	Internal		
	sqrt	Internal		
	log2	Internal		
	log2	Internal		
	log10	Internal		
	log10	Internal		
	log256	Internal		
	log256	Internal		
SafeMathUpgradeable	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	OwnableUpgradeable, ReentrancyGuardUpgradeable		
	initialize	Public	✓	initializer
	getVestingSchedulesCountByBeneficiary	External		-
	getVestingIdAtIndex	External		-
	getVestingScheduleByAddressAndIndex	External		-
	getVestingSchedulesTotalAmount	External		-
	setCrowdsaleAddress	External	✓	-
	getToken	External		-
	createVestingSchedule	Public	✓	onlyCrowdsaleOrOwner
	revoke	Public	✓	onlyOwner onlyIfVestingScheduleNotRevoked
	withdraw	Public	✓	nonReentrant onlyOwner
	releaseTokenForTGE	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		-

Contract Flow



Domain Info

Domain Name	payme.games
Registry Domain ID	29f4ee9286e043058b41ccc27375747f-DONUTS
Creation Date	2021-01-06T13:00:37Z
Updated Date	2022-08-05T11:31:27Z
Registry Expiry Date	2023-01-06T13:00:37Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created almost 2 years before the creation of the audit. It will expire in 29 days.

There is no public billing information, the creator is protected by the privacy settings.

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