Cyberscope

Audit Report

GPAY

November 2022

Type	BEP20
Network	BSC
Address	0xcc735d665e27ea480eb47355969a388b0d8a74d7
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BuyBackToken
Compiler Version	v0.8.5+commit.a4f2e591
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xcC735d665E27eA480EB47355969a388b0D8a74D7
Symbol	GPAY
Decimals	18
Total Supply	21,000,000
Domain	gpaycoins.com

Source Files

Filename	**SHA256**
contract.sol	9f327910e95356213f9284e904e5a3c5bc60e0983cfe4adc609228a0a4a3e0dc

Audit Updates

Initial Audit	7th November 2022 https://github.com/cyberscope-io/audits/tree/main/gpay/v1/audit.pdf
New Iteration	12th November 2022

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🟠	ST	Stops Transactions	Unresolved
🔵	OCTD	Transfers Contract's Tokens	Passed
🔵	OTUT	Transfers User's Tokens	Passed
🔴	ELFM	Exceeds Fees Limit	Unresolved
🔵	ULTW	Transfers Liquidity to Team Wallet	Passed
🔵	MT	Mints Tokens	Passed
🔵	BT	Burns Tokens	Passed
🔵	BC	Blacklists Addresses	Passed

ST - Stops Transactions

Criticality	medium
Location	contract.sol#L674
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users excluding the owner. The owner may take advantage of it by setting the `_maxTxAmount` to zero.

```
if(from != owner() && to != owner()) {
    require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
}
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ELFM - Exceeds Fees Limit

Criticality	critical
Location	contract.sol#L932
Status	Unresolved

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setTaxFee` function with a high percentage value.

```
function setTaxFee(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor / Informative

Severity	Code	Description	Status
🔴	ZD	Zero Division	Unresolved
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved
⚪	L07	Missing Events Arithmetic	Unresolved
⚪	L09	Dead Code Elimination	Unresolved
⚪	L13	Divide before Multiply Operation	Unresolved

ZD - Zero Division

Criticality	critical
Location	contract.sol#L714
Status	Unresolved

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert. This may happen following these steps:

1. presale(true);
2. setSwapAndLiquifyEnabled(true);
3. Transfer an amount when the contract contains more than minimumTokensBeforeSwap tokens.

```
transferToAddressETH(dappbuilderAddress,
transferredBalance.div(_liquidityFee).mul(dappbuilderFee));
```

Recommendation

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L968,951,246,262,460,245,446,936,887,973,959,283,893,963,945
Status	Unresolved

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_enabled
_minimumTokensBeforeSwap
PERMIT_TYPEHASH
MINIMUM_LIQUIDITY
_maxTxAmount
DOMAIN_SEPARATOR
_taxFee
_buybackFee
_amount
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L945,955,951,941,936,932
Status	Unresolved

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
marketingFee = _marketingFee
buyBackUpperLimit = buyBackLimit
minimumTokensBeforeSwap = _minimumTokensBeforeSwap
_maxTxAmount = maxTxAmount
buybackFee = _buybackFee
_taxFee = taxFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L107,124,763,128,133,96,120,116
Status	Unresolved

Description

Functions that are not used in the contract, and make the code's size bigger.

```
sendValue
functionCallWithValue
addLiquidity
_functionCallWithValue
isContract
functionCall
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L498,706
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
_maxTxAmount = _tTotal.div(1000).mul(3)
transferToAddressETH(dappbuilderAddress,transferredBalance.div(_liquidityFee).mu
l(dappbuilderFee))
minimumTokensBeforeSwap = _tTotal.div(10000).mul(2)
transferToAddressETH(marketingAddress,transferredBalance.div(_liquidityFee).mul(
marketingFee.sub(dappbuilderFee)))
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-

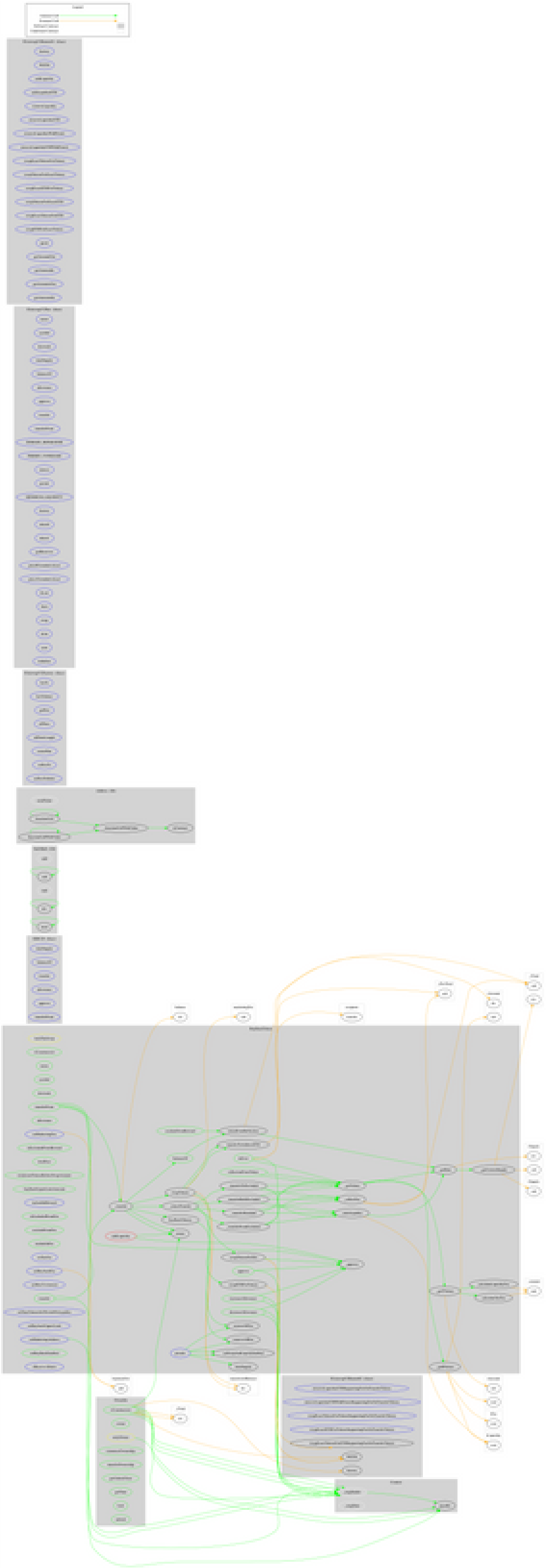
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-

	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
BuyBackToken	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	minimumTokensBeforeSwapAmount	Public		-
	buyBackUpperLimitAmount	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner

	includeInReward	External	✓	onlyOwner
	_approve	Private	✓	
	_transfer	Private	✓	
	swapTokens	Private	✓	lockTheSwap
	buyBackTokens	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	swapETHForTokens	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setTaxFee	External	✓	onlyOwner
	setBuybackFee	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	setBuybackUpperLimit	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner

	setBuyBackEnabled	Public	✓	onlyOwner
	presale	External	✓	onlyOwner
	transferToAddressETH	Private	✓	
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	gpaycoins.com
Registry Domain ID	5839857
Creation Date	2021-10-21T08:24:01Z
Updated Date	2022-11-03T00:56:13Z
Registry Expiry Date	2023-10-21T08:24:01Z
Registrar WHOIS Server	whois.bluehost.com
Registrar URL	http://www.bluehost.com/
Registrar	FastDomain Inc.
Registrar IANA ID	1154

The domain was created about 1 year before the creation of the audit. It will expire in 11 months.

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

Summary

There are some functions that can be abused by the owner like stopping transactions and manipulating fees. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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