Cyberscope

Audit Report

CyberHorseCoin

May 2024

Network	BSC
Address	0xB9C6db0f4a81cBCE412d3E4A2A0A4779F5704c9b
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	GO	Gas Optimization	Unresolved
●	PLPI	Potential Liquidity Provision Inadequacy	Unresolved
●	L17	Usage of Solidity Assembly	Unresolved

Table of Contents

Analysis 1
Diagnostics 2
Table of Contents 3
Review 4
Audit Updates 4
Source Files 4
Findings Breakdown 6
GO - Gas Optimization 7
Description 7
Recommendation 8
PLPI - Potential Liquidity Provision Inadequacy 9
Description 9
Recommendation 9
L17 - Usage of Solidity Assembly 10
Description 10
Recommendation 10
Functions Analysis 11
Inheritance Graph 13
Flow Graph 14
Summary 15
Disclaimer 16
About Cyberscope 17

Review

Contract Name	CHorse
Compiler Version	v0.8.20+commit.a1b79de6
Optimization	9999 runs
Explorer	https://bscscan.com/address/0xb9c6db0f4a81cbce412d3e4a2a0a4779f5704c9b
Address	0xb9c6db0f4a81cbce412d3e4a2a0a4779f5704c9b
Network	BSC
Symbol	CHorse
Decimals	18
Total Supply	10,000,000
Badge Eligibility	Yes

Audit Updates

Initial Audit	18 May 2024

Source Files

Filename	SHA256
src/smartContract/SwapHelper.sol	2f9a019606bebddf4b5dd9f0b29e47519bb92642a9d3a0b79cf3ac9c88e9a450
src/smartContract/GasHelper.sol	6aaf10cfed715b2da4a963289ccc4d902d154c3b27f6eec9ce977f532264ba86

src/smartContract/Events.sol	1e62a06260c92059257ddad9de7051ab30acc9a0c69d38a7a13001c60ff79be7
src/smartContract/Errors.sol	bde04e5bceb6159b2242340ccd8571842dcbc8c35c705850cdb8d93914186e66
src/smartContract/CHorse.sol	280ed2ba283d65f7ba991fece2f9edab6bea67f74b8ba0fd0685fe1dc10daa5c

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	3

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	3	0	0	0

GO - Gas Optimization

Criticality	Minor / Informative
Location	CHorse.sol#L51,57
Status	Unresolved

Description

Gas optimization refers to the process of reducing the amount of gas required to execute a transaction. Gas is the unit of measurement used to calculate the fees paid to miners for including a transaction in a block on the blockchain.

The contract modifies the state of certain variables when the provided argument is different than their current state. However, in the case where the argument matches the currect state of the variable, the contract will not modify the state but the caller of the function will still be charged with gas.

```
function updateExceptFeeWallet(address target, bool status) external
onlyOwner {
  if (exceptFeeWallets[target] == status) return;
  exceptFeeWallets[target] = status;
  emit ExceptFeeWalletsUpdated(target, status);
}
function updateLiquidityWallet(address target, bool status) external
onlyOwner {
  if (liquidityWallets[target] == status) return;
  liquidityWallets[target] = status;
  emit LiquidityWalletsUpdated(target, status);
}
```

Recommendation

The team is advised to take these segments into consideration and rewrite them so the runtime will be more performant. That way it will improve the efficiency and performance of the source code and reduce the cost of executing it.

The contract could modify these segments to use the `require` function provided by Solidity. By doing so, if the condition is not met, the transaction will revert and the caller will not be charged for executing the function.

PLPI - Potential Liquidity Provision Inadequacy

Criticality	Minor / Informative
Location	CHorse.sol#L103,105
Status	Unresolved

Description

The contract operates under the assumption that liquidity is consistently provided to the pair between the contract's token and the native currency. However, there is a possibility that liquidity is provided to a different pair. This inadequacy in liquidity provision in the main pair could expose the contract to risks. Specifically, during eligible transactions, where the contract attempts to swap tokens with the main pair, a failure may occur if liquidity has been added to a pair other than the primary one. Consequently, transactions triggering the swap functionality will result in a revert.

```
swapToken(liquidityPoolLocal, wbnbAmount, 0, swapHelper);
swapToken(liquidityPoolLocal, 0, wbnbAmount, swapHelper);
```

Recommendation

The team is advised to implement a runtime mechanism to check if the pair has adequate liquidity provisions. This feature allows the contract to omit token swaps if the pair does not have adequate liquidity provisions, significantly minimizing the risk of potential failures.

Furthermore, the team could ensure the contract has the capability to switch its active pair in case liquidity is added to another pair.

Additionally, the contract could be designed to tolerate potential reverts from the swap functionality, especially when it is a part of the main transfer flow. This can be achieved by executing the contract's token swaps in a non-reversible manner, thereby ensuring a more resilient and predictable operation.

L17 - Usage of Solidity Assembly

Criticality	Minor / Informative
Location	GasHelper.sol#L20,33,49
Status	Unresolved

Description

Using assembly can be useful for optimizing code, but it can also be error-prone. It's important to carefully test and debug assembly code to ensure that it is correct and does not contain any errors.

Some common types of errors that can occur when using assembly in Solidity include Syntax, Type, Out-of-bounds, Stack, and Revert.

```
assembly {
      let emptyPointer := mload(0x40)
      mstore(emptyPointer,
0x0dfe1681000000000000000000000000000000000000000000000000000000000000000)
      failed := iszero(staticcall(gas(), pair, emptyPointer, 0x04,
emptyPointer, 0x20))
      token0 := mload(emptyPointer)
    }

...
```

Recommendation

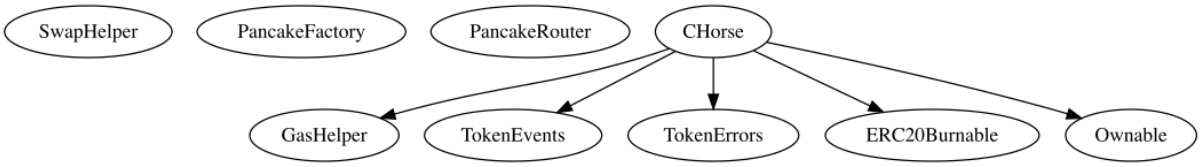
It is recommended to use assembly sparingly and only when necessary, as it can be difficult to read and understand compared to Solidity code.

Functions Analysis

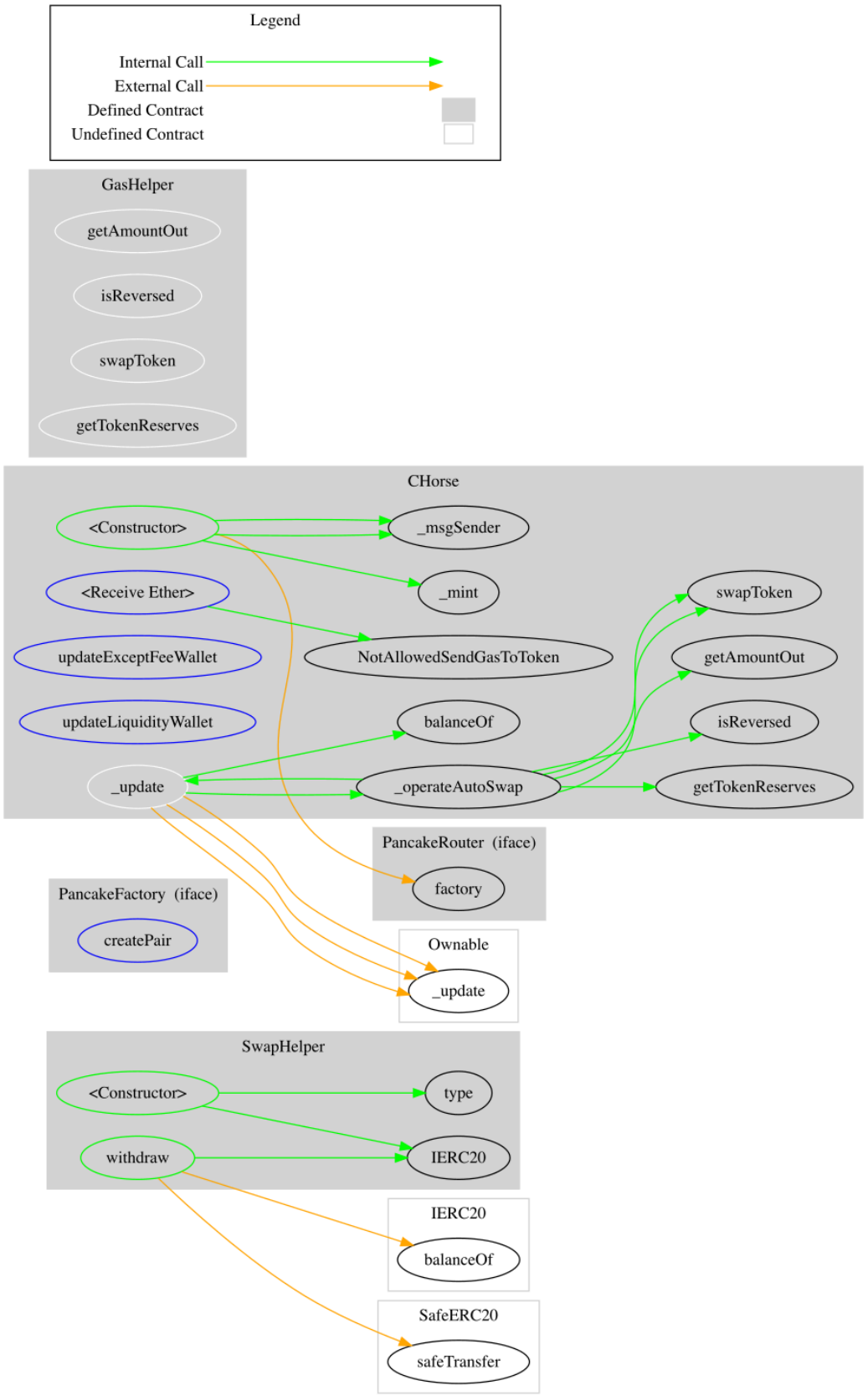
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SwapHelper	Implementation			
		Public	✓	-
	withdraw	Public	✓	-
GasHelper	Implementation			
	getAmountOut	Internal		
	isReversed	Internal		
	swapToken	Internal	✓	
	getTokenReserves	Internal		
TokenEvents	Interface			
TokenErrors	Interface			
CHorse	Implementation	ERC20Burnable, GasHelper, TokenErrors, TokenEvents, Ownable		
		Public	✓	ERC20 Ownable
		External	Payable	-

	updateExceptFeeWallet	External	✓	onlyOwner
	updateLiquidityWallet	External	✓	onlyOwner
	_update	Internal	✓	
	_operateAutoSwap	Private	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
GasHelper
getAmountOut
isReversed
swapToken
getTokenReserves
CHorse
<Constructor>
_msgSender
_mint
<Receive Ether>
NotAllowedSendGasToToken
updateExceptFeeWallet
updateLiquidityWallet
balanceOf
_update
_operateAutoSwap
swapToken
getAmountOut
isReversed
getTokenReserves
PancakeRouter (iface)
factory
PancakeFactory (iface)
createPair
Ownable
_update
SwapHelper
<Constructor>
type
withdraw
IERC20
IERC20
balanceOf
SafeERC20
safeTransfer

Summary

CyberHorseCoin contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. CyberHorseCoin is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. The fees are locked at 3% for both buys and sales.

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