Cyberscope

Audit Report
Lambro

May 2024

Network	BSC
Address	0xA7cD7b237A6e928cD507e5bE4AEe953d7482E80E
Audited by	© cyberscope

Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🔴	ST	Stops Transactions	Unresolved
🔵	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
●	OCTD	Transfers Contract's Tokens	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Analysis **1**
Diagnostics **2**
Table of Contents **3**
Review **4**
Audit Updates 4
Source Files 4
Findings Breakdown **6**
ST - Stops Transactions 7
Description 7
Recommendation 8
OCTD - Transfers Contract's Tokens 9
Description 9
Recommendation 10
L19 - Stable Compiler Version 11
Description 11
Recommendation 11
Functions Analysis **12**
Inheritance Graph **13**
Flow Graph **14**
Summary **15**
Disclaimer **16**
About Cyberscope **17**

Review

Contract Name	Lambro
Compiler Version	v0.8.23+commit.f704f362
Optimization	200 runs
Explorer	https://bscscan.com/address/0xa7cd7b237a6e928cd507e5be4aee953d7482e80e
Address	0xa7cd7b237a6e928cd507e5be4aee953d7482e80e
Network	BSC
Symbol	LAMBRO
Decimals	18
Total Supply	1,000,000,000
Badge Eligibility	Yes

Audit Updates

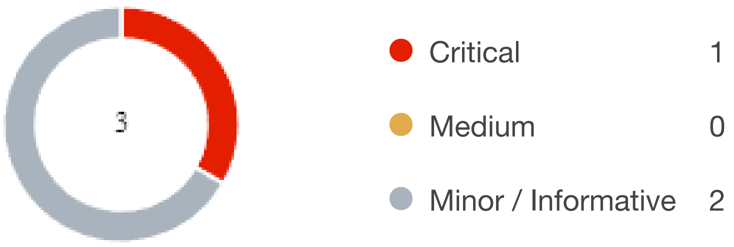
Initial Audit	04 May 2024

Source Files

Filename	SHA256
Lambro.sol	c5ba905df86af85cfdde97020da2bc62a423669dc465f14f0642852cb562505a
@openzeppelin/contracts/utils/Context.sol	847fda5460fee70f56f4200f59b82ae622bb03c79c77e67af010e31b7e2cc5b6

@openzeppelin/contracts/utils/Address.sol	b3710b1712637eb8c0df81912da3450da6ff67b0b3ed18146b033ed15b1aa3b9
@openzeppelin/contracts/token/ERC20/IERC20.sol	6f2faae462e286e24e091d7718575179644dc60e79936ef0c92e2d1ab3ca3cee
@openzeppelin/contracts/token/ERC20/ERC20.sol	2d874da1c1478ed22a2d30dcf1a6ec0d09a13f897ca680d55fb49fbcc0e0c5b1
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	471157c89111d7b9eab456b53ebe9042bc69504a64cb5cc980d38da9103379ae
@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	912509e0e9bf74e0f8a8c92d031b5b26d2d35c6d4abf3f56251be1ea9ca946bf
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	1d079c20a192a135308e99fa5515c27acfbb071e6cdb0913b13634e630865939
@openzeppelin/contracts/token/ERC20/extensions/ERC20Capped.sol	cb15f210495f2119cfec53e32738ddb23469d414c45a2d7444c2181a9940bbed
@openzeppelin/contracts/security/ReentrancyGuard.sol	fa97ea556c990ee44f2ef4c80d4ef7d0af3f5f9b33a0214291140688106f5a9
@openzeppelin/contracts/security/Pausable.sol	2072248d2f79e661c149fd6a6593a8a3f038466557c9b75e50e0b001bcb5cf97
@openzeppelin/contracts/interfaces/draft-IERC6093.sol	4aea87243e6de38804bf8737bf86f750443d3b5e63dd0fd0b7ad92f77cdbd3e3
@openzeppelin/contracts/access/Ownable.sol	38578bd71c0a909840e67202db527cc6b4e6b437e0f39f0c909da32c1e30cb81

Findings Breakdown



Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	1	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	2	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	Lambro.sol#L67
Status	Unresolved

Description

The contract owner has the authority to stop the sales for all users. The owner may take advantage of it by calling the `pause` function.

```
function pause() public onlyOwner {
    _pause();
    emit TokenPaused(_msgSender());
}

function _update(address from, address to, uint256 amount)
    internal
    override(ERC20, ERC20Capped)
    whenNotPaused
{
    super._update(from, to, amount);
}
```

Recommendation

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

OCTD - Transfers Contract's Tokens

Criticality	Minor / Informative
Location	Lambro.sol#L102
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `retrieveTokens` function.

```
function retrieveTokens(address tokenAddress, address to,
uint256 amount) external onlyOwner {
    require(tokenAddress != address(0), "Lambro: retrieve from
zero address");
    require(to != address(0), "Lambro: transfer to zero
address");
    require(amount > 0, "Lambro: amount must be greater than
zero");

    IERC20 token = IERC20(tokenAddress);
    token.safeTransfer(to, amount);
    emit TokensRetrieved(tokenAddress, to, amount);
}
```

Recommendation

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	Lambro.sol#L1
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```
pragma solidity ^0.8.23;
```

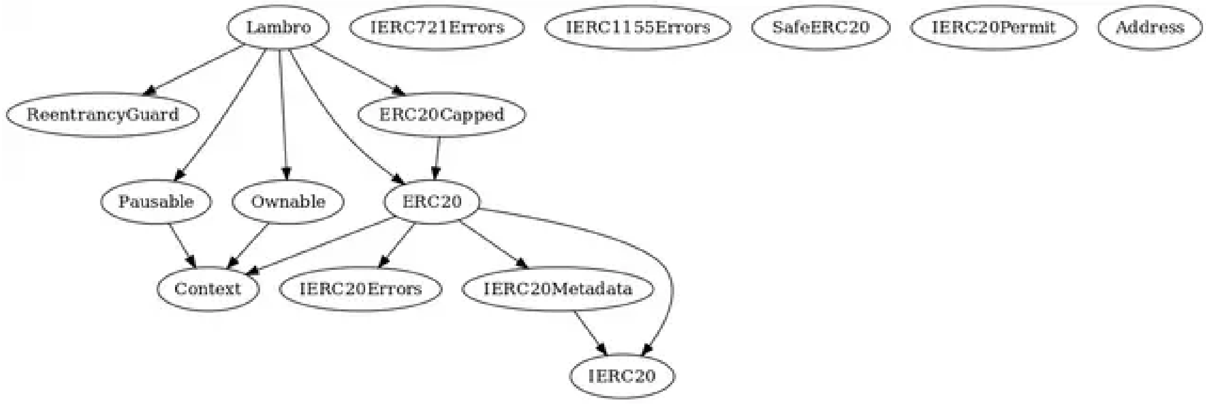
Recommendation

The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

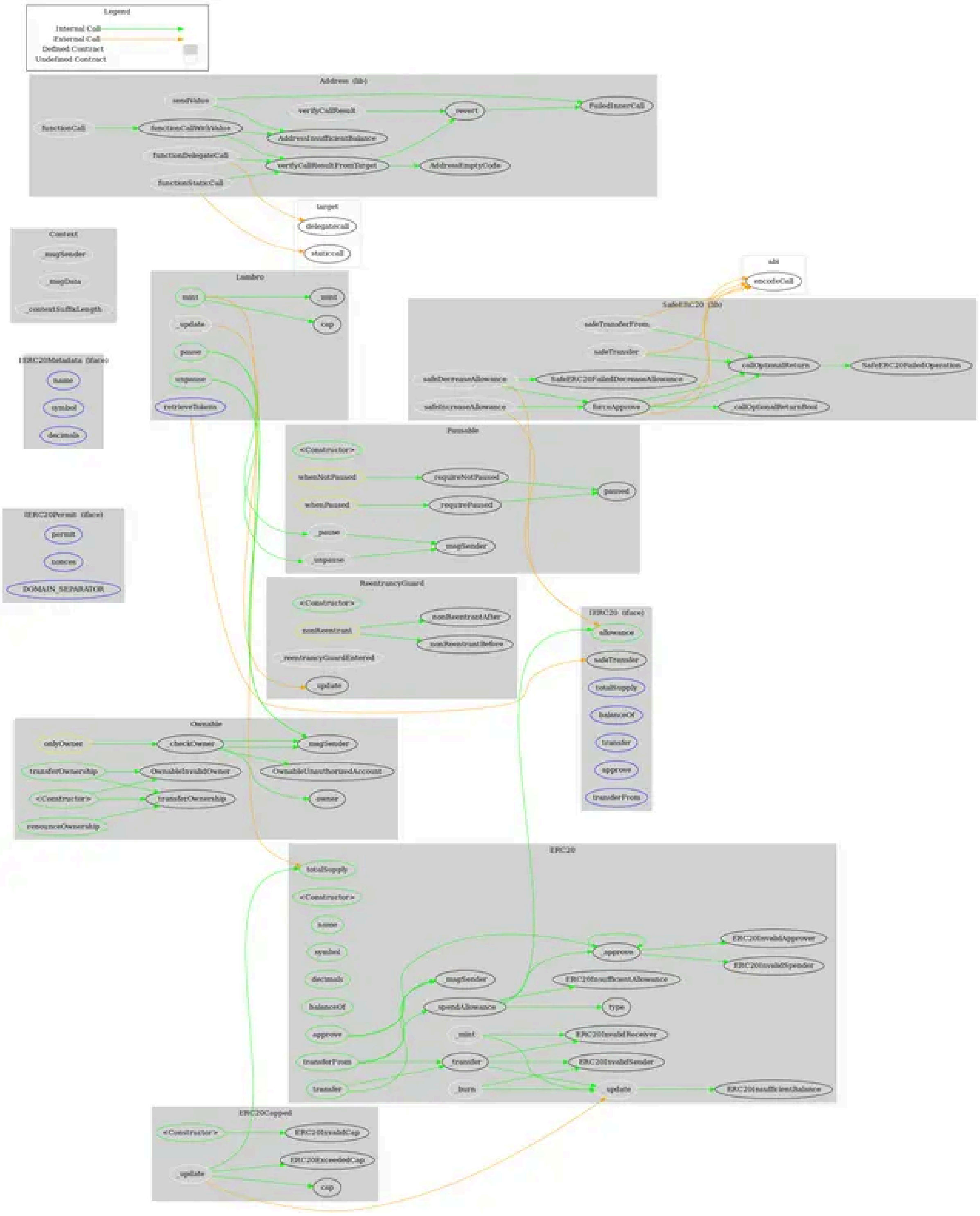
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Lambro	Implementation	ERC20, ERC20Capped, Pausable, Ownable, ReentrancyGuard		
		Public	✓	ERC20 ERC20Capped Ownable
	mint	Public	✓	onlyOwner
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	_update	Internal	✓	whenNotPaused
	retrieveTokens	External	✓	onlyOwner

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Address (lib)
sendValue
verifyCallResult
revert
FailedInnerCall
functionCall
functionCallWithValue
AddressInsufficientBalance
functionDelegateCall
verifyCallResultFromTarget
AddressEmptyCode
functionStaticCall
target
delegatecall
staticcall
Context
_msgSender
_msgData
_contextSuffixLength
IERC20Metadata (iface)
name
symbol
decimals
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR
Lambro
mint
_mint
cap
_update
pause
unpause
retrieveTokens
abi
encodeCall
SafeERC20 (lib)
safeTransferFrom
safeTransfer
callOptionalReturn
SafeERC20FailedOperation
safeDecreaseAllowance
SafeERC20FailedDecreaseAllowance
safeIncreaseAllowance
forceApprove
callOptionalReturnBool
Pausable
<Constructor>
whenNotPaused
_requireNotPaused
whenPaused
_requirePaused
paused
_pause
_msgSender
_unpause
ReentrancyGuard
<Constructor>
_nonReentrantAfter
nonReentrant
_nonReentrantBefore
_reentrancyGuardEntered
_update
IERC20 (iface)
allowance
safeTransfer
totalSupply
balanceOf
transfer
approve
transferFrom
Ownable
onlyOwner
_checkOwner
_msgSender
transferOwnership
OwnableInvalidOwner
OwnableUnauthorizedAccount
<Constructor>
_transferOwnership
owner
renounceOwnership
ERC20
totalSupply
<Constructor>
name
symbol
decimals
balanceOf
approve
transferFrom
transfer
_approve
ERC20InvalidApprover
ERC20InvalidSpender
_msgSender
ERC20InsufficientAllowance
_spendAllowance
type
_mint
ERC20InvalidReceiver
_transfer
ERC20InvalidSender
_burn
_update
ERC20InsufficientBalance
ERC20Capped
<Constructor>
ERC20InvalidCap
ERC20ExceededCap
_update
cap

Summary

Lambro contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. There are some functions that can be abused by the owner like stop transactions. 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