Cyberscope

Audit Report

ZETA

September 2023

Network	Goerli
Address	0x4d2681d8585e4471c587f1fce77a50bc89775152
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
	DDP	Decimal Division Precision	Unresolved
	MDIEE	Missing Data in Event Emission	Unresolved
	RSW	Redundant Storage Writes	Unresolved
	PTRP	Potential Transfer Revert Propagation	Unresolved
	FSA	Fixed Swap Address	Unresolved
	IDI	Immutable Declaration Improvement	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L13	Divide before Multiply Operation	Unresolved

Table of Contents

Review

Explorer	https://goerli.etherscan.io/address/0x4d2681d8585e4471c587f1fce77a50bc89775152

Audit Updates

Initial Audit	23 Sep 2023

Source Files

Filename	SHA256
stoicDAO.sol	60e18ad7967bb2a07ea04c7624ede732dcc5ff12a54e1b9b1439e1c8baa99e56

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	9

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	9	0	0	0

ST - Stops Transactions

Criticality	Minor / Informative
Location	stoicDAO.sol#L302
Status	Unresolved

Description

The transactions are initially disabled for all users excluding the authorized addresses. The owner can enable the transactions for all users. Once the transactions are enable the owner will not be able to disable them again.

```
if(!excludedFromFees[sender] && !excludedFromFees[recipient] &&
!swapping){
    require(launched, "Trading not active yet");
    require(amount <= maxTxAmount, "You are exceeding
maxTxAmount");
    if(recipient != pair){
        require(balanceOf(recipient) + amount <=
maxWalletAmount, "You are exceeding maxWalletAmount");
    }
}
```

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. Some suggestions are:

- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

DDP - Decimal Division Precision

Criticality	Minor / Informative
Location	stoicDAO.sol#L347
Status	Unresolved

Description

Division of decimal (fixed point) numbers can result in rounding errors due to the way that division is implemented in Solidity. Thus, it may produce issues with precise calculations with decimal numbers.

Solidity represents decimal numbers as integers, with the decimal point implied by the number of decimal places specified in the type (e.g. decimal with 18 decimal places). When a division is performed with decimal numbers, the result is also represented as an integer, with the decimal point implied by the number of decimal places in the type. This can lead to rounding errors, as the result may not be able to be accurately represented as an integer with the specified number of decimal places.

Hence, the splitted shares will not have the exact precision and some funds may not be calculated as expected.

```
uint256 deltaBalance = address(this).balance - initialBalance;
uint256 unitBalance= deltaBalance / (denominator -
sellTaxes.liquidity);
uint256 ethToAddLiquidityWith = unitBalance * sellTaxes.liquidity;

if(tokensToAddLiquidityWith > 0 && ethToAddLiquidityWith > 0){
    // Add liquidity to dex
    addLiquidity(tokensToAddLiquidityWith, ethToAddLiquidityWith);
}

uint256 marketingAmt = unitBalance * 2 * sellTaxes.marketing;
if(marketingAmt > 0){
    payable(marketingWallet).sendValue(marketingAmt);
}

uint256 developmentAmt = unitBalance * 2 * sellTaxes.development;
if(developmentAmt > 0){
    payable(developmentWallet).sendValue(developmentAmt);
}
uint256 stoicFundAmt = unitBalance * 2 * sellTaxes.stoicFund;
if(stoicFundAmt > 0){
    payable(stoicFundWallet).sendValue(stoicFundAmt);
}

uint256 incubatorAmt = unitBalance * 2 * sellTaxes.incubator;
if(developmentAmt > 0){
    payable(incubatorWallet).sendValue(incubatorAmt);
}
```

Recommendation

The team is advised to take into consideration the rounding results that are produced from the solidity calculations. The contract could calculate the subtraction of the divided funds in the last calculation in order to avoid the division rounding issue.

MDIEE - Missing Data in Event Emission

Criticality	Minor / Informative
Location	stoicDAO.sol#L405,412,419,426,434,441,448,455,462,467,479,492
Status	Unresolved

Description

The contract emits events that lack essential data, making it challenging for users and external applications to track and interpret contract activities accurately. Emitting events for significant actions with essential data is important as it allows external parties, such as wallets or dApps, to track and monitor the activity on the contract. Without the necessary data, it may be difficult for external parties to accurately determine the current state of the contract.

```
emit MaxWalletAmountUpdated();
emit MaxTxAmountUpdated();
....
```

Recommendation

It is recommended to include all relevant details such as the user's address and the nature of the action taken, each time a significant action is taking place within the contract. By doing so, the contract will be more transparent and easily auditable by external parties. It will also help prevent potential issues or disputes that may arise in the future.

RSW - Redundant Storage Writes

Criticality	Minor / Informative
Location	stoicDAO.sol#L403,437,444,451,458,465
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The contract updates variable even if its current state is the same as the one passed as an argument. As a result, the contract performs redundant storage writes.

```
function setSwapEnabled(bool state) external onlyOwner { // to be used
only in case of dire emergency
    swapEnabled = state;
    emit SwapEnabled();
}

...
```

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.

PTRP - Potential Transfer Revert Propagation

Criticality	Minor / Informative
Location	stoicDAO.sol#L356
Status	Unresolved

Description

The contract sends funds to a `marketingWallet` , `developmentWallet` , `stoicFundWallet` , and an `incubatorWallet` as part of the transfer flow. This address can either be a wallet address or a contract. If the address belongs to a contract then it may revert from incoming payment. As a result, the error will propagate to the token's contract and revert the transfer.

```
uint256 marketingAmt = unitBalance * 2 * sellTaxes.marketing;
if(marketingAmt > 0){
    payable(marketingWallet).sendValue(marketingAmt);
}

uint256 developmentAmt = unitBalance * 2 * sellTaxes.development;
if(developmentAmt > 0){
    payable(developmentWallet).sendValue(developmentAmt);
}
uint256 stoicFundAmt = unitBalance * 2 * sellTaxes.stoicFund;
if(stoicFundAmt > 0){
    payable(stoicFundWallet).sendValue(stoicFundAmt);
}

uint256 incubatorAmt = unitBalance * 2 * sellTaxes.incubator;
if(developmentAmt > 0){
    payable(incubatorWallet).sendValue(incubatorAmt);
}
```

Recommendation

The contract should tolerate the potential revert from the underlying contracts when the interaction is part of the main transfer flow. This could be achieved by not allowing set contract addresses or by sending the funds in a non-revertable way.

FSA - Fixed Swap Address

Criticality	Minor / Informative
Location	stoicDAO.sol#L281
Status	Unresolved

Description

The swap address is assigned once and it can not be changed. It is a common practice in decentralized exchanges to create new swap versions. A contract that cannot change the swap address may not be able to catch up to the upgrade. As a result, the contract will not be able to migrate to a new liquidity pool pair or decentralized exchange.

```
constructor() ERC20("stoicDAO", "ZETA") {
    _mint(msg.sender, 1000000000 * 10 ** decimals());
    excludedFromFees[msg.sender] = true;

    IRouter _router =
IRouter(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D);
    address _pair = IFactory(_router.factory())
        .createPair(address(this), _router.WETH());

    router = _router;
    ...
}
```

Recommendation

The team is advised to add the ability to change the pair and router address in order to cover potential liquidity pool migrations. It would be better to support multiple pair addresses so the token will be able to have the same behavior in all the decentralized liquidity pairs.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	stoicDAO.sol#L289,290
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
router
pair
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	stoicDAO.sol#L49,51,205,223,408,422,429,465,482
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
mapping (address => uint256) internal _balances
mapping (address => mapping (address => uint256)) internal _allowances
function WETH() external pure returns (address);

...
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

Find more information on the Solidity documentation
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L13 - Divide before Multiply Operation

Criticality	Minor / Informative
Location	stoicDAO.sol#L316,327,348,349,356,361,365,370
Status	Unresolved

Description

It is important to be aware of the order of operations when performing arithmetic calculations. This is especially important when working with large numbers, as the order of operations can affect the final result of the calculation. Performing divisions before multiplications may cause loss of prediction.

```
uint256 unitBalance= deltaBalance / (denominator - sellTaxes.liquidity)
uint256 marketingAmt = unitBalance * 2 * sellTaxes.marketing
```

Recommendation

To avoid this issue, it is recommended to carefully consider the order of operations when performing arithmetic calculations in Solidity. It's generally a good idea to use parentheses to specify the order of operations. The basic rule is that the multiplications should be prior to the divisions.

Functions Analysis

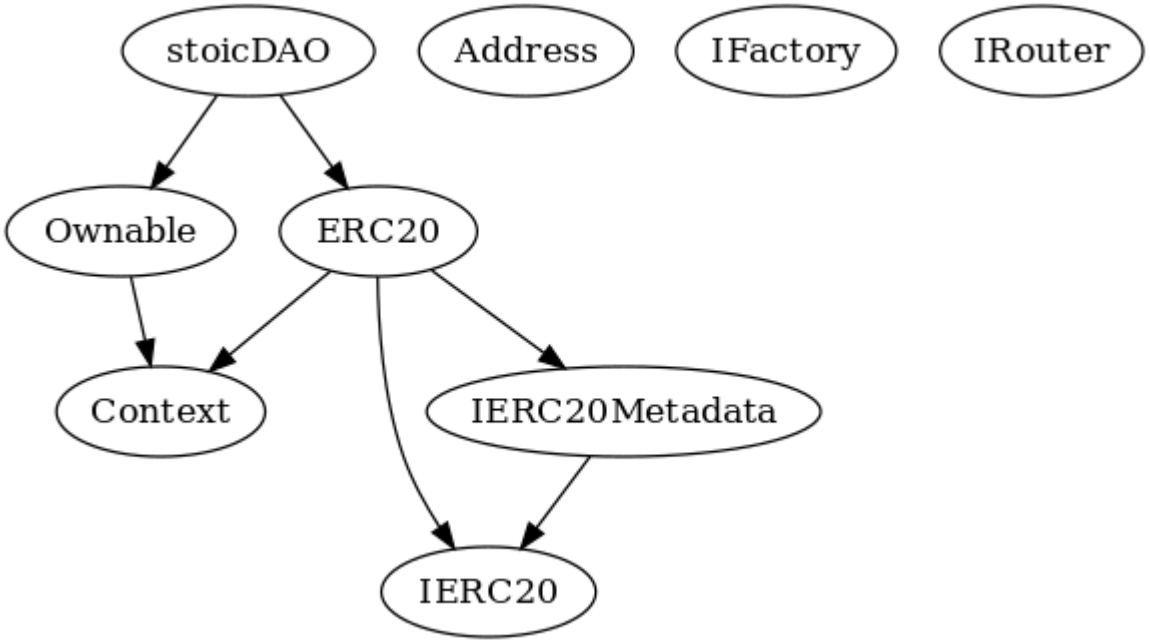
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-

ERC20	Implementation	Context, IERC20, IERC20Metadata		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
Address	Library			
	sendValue	Internal	✓	
Ownable	Implementation	Context		

		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
IFactory	Interface			
	createPair	External	✓	-
IRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
stoicDAO	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	_transfer	Internal	✓	
	swapForFees	Private	✓	inSwap
	swapTokensForETH	Private	✓	
	addLiquidity	Private	✓	
	setSwapEnabled	External	✓	onlyOwner
	setSwapThreshold	External	✓	onlyOwner

	launch	External	✓	onlyOwner
	setBuyTaxes	External	✓	onlyOwner
	setSellTaxes	External	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setDevelopmentWallet	External	✓	onlyOwner
	setStoicFundWallet	External	✓	onlyOwner
	setIncubatorWallet	External	✓	onlyOwner
	setExcludedFromFees	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setMaxWalletAmount	External	✓	onlyOwner
	withdrawStuckTokens	External	✓	onlyOwner
	clearStuckEthers	External	✓	onlyOwner
		External	Payable	-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IFactory (iface)
createPair
Address (lib)
sendValue
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transfer
allowance
transferFrom
increaseAllowance
decreaseAllowance
approve
_transfer
_msgSender
_mint
_approve
_beforeTokenTransfer
Ownable
transfer
onlyOwner
transferOwnership
renounceOwnership
<Constructor>
owner
_msgSender
_setOwner
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
stoicDAO
inSwap
<Constructor>
_transfer
decimals
setSwapEnabled
setSwapThreshold
launch
setBuyTaxes
setSellTaxes
setMarketingWallet
setDevelopmentWallet
setStoicFundWallet
setIncubatorWallet
setExcludedFromFees
setMaxTxAmount
setMaxWalletAmount
withdrawStuckTokens
swapForFees
<Receive Ether>
clearStuckEthers
balanceOf
addLiquidity
swapTokensForETH
payable
IRouter (iface)
factory
WETH
addLiquidityETH
swapExactTokensForETHSupportingFeeOnTransferTokens

Summary

ZETA 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 stopping 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. There is also a limit of max 5% fees.

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