Cyberscope

Audit Report

Insignis

April 2022

Type	ERC20
Network	ONE
Address	0x140320a14b1422ca36b45a41c46ec8179298ceb5
Audited by	© cyberscope

Table of Contents

Contract Review

Insignis

Contract Name	Insignis
Compiler Version	0.7.6+commit.7338295f
Optimization	No
Licence	
Explorer	https://explorer.harmony.one/address/0x140320a14b1422ca36b45a41c46ec8179298ceb5
Symbol	INSIG
Decimals	6
Total Supply	9314636376
Domain	insignis.finance

WINSIG

Contract Name	WINSIG
Compiler Version	0.7.6+commit.7338295f
Optimization	No
Licence	
Explorer	https://explorer.harmony.one/address/0x0aeec7718baefc4a4e322a35c387d75b9c2c911e

Symbol	INSIG
Decimals	6
Total Supply	458293651076
Domain	insignis.finance

Source Files

Filename	**SHA256**
Insignis	0705469783c996d5e89d3dc16d7c7e601ddcacb468387c226a3168d6bd1e5195
WINSIG	93d170927f8ef1788c50c78ec192f5d08689161ff3195bdece1c92827ebca3e3

Audit Updates

Initial Audit	1st April 2022
Corrected	5th April 2022

Contract Analysis

Wrapped Contract

The WINSIG contract wraps the INSIG token according to a specific ratio. This ratio cannot be changed. That means that the team should be extra careful if it is going to trade the WINSIG token as well. When the WINSIG price differs from the INSIG price, then users will be able to buy the first one at a low price, convert it, and sell the other one at a high price. The WINSIG/INSIG should be stable according to the “index” ratio.

- wrap function converts INSIG to WINSIG.
- unwrap function converters WINSIG to INSIG.

Critical Medium Minor Pass

Severity	Code	Description	Resolved
●	ST	Contract Owner is not able to stop or pause transactions	
●	OCTD	Contract Owner is not able to transfer tokens from specific address	
●	OTUT	Owner Transfer User's Tokens	
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)	
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent	
●	MT	Contract Owner is not able to mint new tokens	**Resolved**
●	BT	Contract Owner is not able to burn tokens from specific wallet	
●	BC	Contract Owner is not able to blacklist wallets from selling	

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L726
Status	Resolved

Description

The wrapped role has the authority to mint tokens. The wrapped role may take advantage of it by calling the `mint` function. As a result the contract tokens will be highly inflated.

```
function mint(address recipient, uint256 amount) external onlyWrapped {
    // Only WINSIG can mint tokens to provide additional rewards.
    _totalSupply = _totalSupply.add(uint256(amount));

    uint256 gonsAmount = amount.mul(_gonsPerFragment);

    if (_totalSupply > MAX_SUPPLY) {
        _totalSupply = MAX_SUPPLY;
    }

    _gonsPerFragment = TOTAL_GONS.div(_totalSupply);
    pairContract.sync();

    _gonBalances[recipient] = _gonBalances[recipient].add(gonsAmount);

    emit Mint(recipient, amount.div(_gonsPerFragment));
}
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Update 5 April

The team has renounced the wrapped role ownership. The wrapped role cannot be configured any more. That means that the mint functionality cannot be called anymore.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	MTS	Manipulate Total Supply
	MC	Missing Check
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L06	Missing Events Access Control
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L13	Divide before Multiply Operation
	L15	Local Scope Variable Shadowing

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L707

Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap

```
function rebase() external isRebaser returns (uint256) {
    require(!inSwap, "Try again");
    _totalSupply = _totalSupply.add(getRebase());

    addEpoch();

    if (_totalSupply > MAX_SUPPLY) {
        _totalSupply = MAX_SUPPLY;
    }

    _gonsPerFragment = TOTAL_GONS.div(_totalSupply);
    pairContract.sync();

    emit Rebase(indexer.index, _totalSupply);
    return _totalSupply;
}
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token's price fluctuations.

MC - Missing Check

Criticality	minor
Location	contract.sol#L2185

Description

According to the wrap functionality, the user can transform INSIG tokens to WINSIG tokens. This is happening in two steps:

1. The user transforms INSIG tokens to the WINSIG contract.
2. The WINSIG contract mints the corresponding WINSIG amount to the user.

According to the IERC20 transferFrom API, the contract should not rely on the external implementation and check if the transferFrom function has successfully proceeded. Otherwise, the user may take tokens without paying the corresponding amount.

```
function wrap(uint256 _amount) external returns (uint256) {
    IERC20(INSIG).transferFrom(msg.sender, address(this), _amount);

    uint256 value = INSIGtoWINSIG(_amount);
    _mint(msg.sender, value);
    return value;
}
```

Recommendation

The contract should check if the transformFrom has been successfully processed. The contract should initially check if the sender’s balance is sufficient for the transaction.

Respectively, the same checks should happen in the unwrap function.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L457,463,469,509,521,528,746,1073,1758,1766 and 10 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
stake
decreaseAllowance
increaseAllowance
transferFrom
approve
allowance
transfer
balanceOf
totalSupply
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L559,571,572,558,2084,2086

Description

Constant state variables should be declared constant to save gas.

```
timelock_duration
_totalSupply
totalFee
targetLiquidityDenominator
targetLiquidity
feeDenominator
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L227,546,1008,1014,1174,1181,1189,1195,1201,1208 and 29 more

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.

```
timelock_percentage
timelock_duration
INSIG
_amount
INSIGtoWINSIG
WINSIGtoINSIG
lock
wl_bought
_isFeeExempt
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L1365,2084

Description

There are segments that contain unused state variables.

```
_totalSupply
MAX_INT256
```

Recommendation

Remove unused state variables.

L06 - Missing Events Access Control

Criticality	minor
Location	contract.sol#L1189,1229,2219

Description

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

```
owner = addr
wrapped = addr
rebaser = _address
```

Recommendation

Emit an event for critical parameter changes.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1243

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.

```
gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L1639,1495,1508,1527,1547,1609,1627,1572,1591,1434 and 8 more

Description

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

```
abs
remove
has
transferToAddressETH
_index
_approve
_setupDecimals
sendValue
isContract
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L800,1229,1307,1336

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
wl[tx.origin] == true
wl_bought[tx.origin] == true
require(bool,string)(wrapped_set == false,Wrapped contract address can only be
defined once.)
require(bool,string)(isTradable(amount) == true,You cannot do more than one
transaction before the launch date.)
```

Recommendation

Remove the equality to the boolean constant.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L857,1155,1243

Description

Performing divisions before multiplications may cause lose of prediction.

```
gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num)
liquidityBalance = _gonBalances[pair].div(_gonsPerFragment)
contractTokenBalance = _gonBalances[address(this)].div(_gonsPerFragment)
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L445,446,447

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
decimals
symbol
name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
Roles	Library			
	add	Internal	✓	
	remove	Internal	✓	
	has	Internal		
InterfaceLP	Interface			
	sync	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-

	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	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	✓	-
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Insignis	Implementation	ERC20Detailed, Ownable		
	<Constructor>	Public	✓	ERC20Detailed
	resetWLSwapLimit	External	✓	onlyOwner
	rebase	External	✓	isRebaser
	mint	External	✓	onlyWrapped
	burn	Public	✓	-
	totalSupply	External		-
	transfer	External	✓	validRecipient
	allowance	External		-
	balanceOf	External		-
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	transferFrom	External	✓	validRecipient
	swapBack	Internal	✓	swapping
	takeFee	Internal	✓	
	decreaseAllowance	External	✓	-

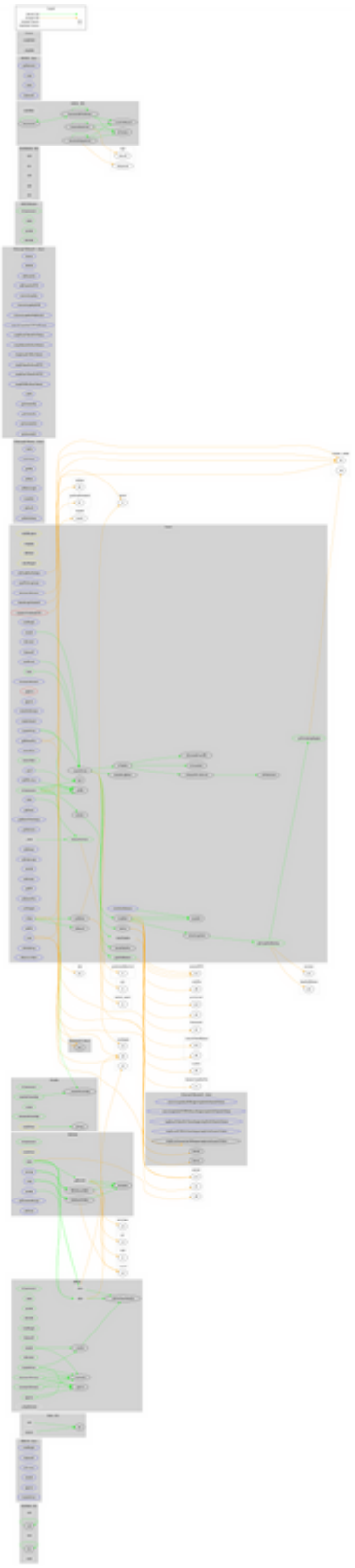
	increaseAllowance	External	✓	-
	_approve	Private	✓	
	approve	External	✓	-
	checkFeeExempt	External		-
	enableTransfer	External	✓	onlyOwner
	shouldTakeFee	Internal		
	shouldSwapBack	Internal		
	sendPresale	External	✓	onlyOwner
	checkSwapThreshold	External		-
	manualSync	External	✓	-
	clearStuckBalance	External	✓	onlyOwner
	rescueToken	Public	✓	onlyOwner
	transferToAddressETH	Private	✓	
	gonsForBalance	Public		-
	balanceForGons	Public		-
	index	External		-
	_index	Internal		
	getRebase	Public		-
	getRebasePerc	External		-
	getEpoch	External		-
	getEpochTimestamp	External		-
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	getDecimals	External		-
	setIndex	Internal	✓	
	setLP	External	✓	onlyOwner
	setRebaser	External	✓	onlyOwner
	setFeeExempt	External	✓	onlyOwner
	setALR	External	✓	onlyOwner
	setTreasury	External	✓	onlyOwner
	setRFV	External	✓	onlyOwner
	setEpochNow	External	✓	onlyOwner
	setWrapped	External	✓	onlyOwner
	setSwapBackSettings	External	✓	onlyOwner
	_addWL	Internal	✓	

	addWL	External	✓	onlyOwner
	addWLArray	External	✓	onlyOwner
	addEpoch	Internal	✓	
	isNotInSwap	External		-
	isOverLiquified	Public		-
	isTradable	Public	✓	-
	isTransacWLAllowed	Public	✓	-
	isLaunched	Public		-
	isWhitelisted	Public		-
	isExcludedFromWL	Public		-
	<Receive Ether>	External	Payable	-
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
IINSIG	Interface			
	getDecimals	External		-
	mint	External	✓	-

	index	External		-
	balanceOf	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20	Implementation	Context, IERC20		
	<Constructor>	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	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
WINSIG	Implementation	ERC20		
	<Constructor>	Public	✓	ERC20
	getReward	Internal		
	stake	Public	✓	-
	unstake	External	✓	-
	wrap	External	✓	-
	unwrap	External	✓	-
	decimaled	Internal		

	setTimelockReward	External	✓	onlyOwner
	setOwner	External	✓	onlyOwner
	WINSIGtoINSIG	Public		-
	INSIGtoWINSIG	Public		-

Contract Flow



Domain Info

Domain Name	insignis.finance
Registry Domain ID	9f1aa3d7edec4ee8892cec7c290d71ba-DONUTS
Creation Date	2022-02-22T22:59:41Z
Updated Date	2022-02-27T23:00:11Z
Registry Expiry Date	2023-02-22T22:59:41Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created about 1 month 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

Insignis is an interesting project that has a friendly and growing community. The Smart Contract analysis reported one critical issue. The wrapped role may mint an arbitrary number of tokens and inflate the contract's balance. We state that the owner privileges are necessary and required for proper protocol operations of the staking contract. Thus, we emphasise the contract owner to be extra careful with the credentials. There is also a limit of max 12% fees in buys and 19% in sales. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Update 5 April

The team has renounced the wrapped role ownership. The wrapped role cannot be configured any more. That means that the mint functionality cannot be called anymore.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io