Cyberscope

Audit Report

Star Link Satellite

June 2022

SHA256 f28412d49068685823c2332b96765d53c3df3d3b859ce77b4e405102632f30dc

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	Deck
Symbol	Deck
Decimals	18
Domain	starlinksats-finance.web.app

Audit Updates

Initial Audit	2nd July 2022
Corrected	

Notes

The Star Link Satellite contract implements a token functionality enriched with a funds distribution mechanism. The functionality of the contract heavily depends on external sources. The contracts that depend on external sources should be extra careful since they could be manipulated. This audit focuses on the Star Link Satellite contract. The auditing of the external sources are out of the scope of the audit.

Source Files

Filename	SHA256
Deck.sol	f28412d49068685823c2332b96765d53c3df3d3b859ce77b4e405102632f30dc
ERC20Upgradeable.sol	74edf948ecc2e71014919516138e880cc1e28c1b617aec6c0be5b52d6063d2b9
GovernableUpgradeable.sol	aa490bccb4952e75c87832ab0711d6278332354b096520db191b9f728a67aec0
IERC20.sol	05c55b7774c01169eff8bdb4da8d6266dc28b5a182c75aa9c82a59d896ebd937
IFeeDistributor.sol	10c61b7d3ff7c007e58973df2d74d6ddafd1c60c6dec3f8b8f69dd4f128087b1
IPriceRegulator.sol	fbb9abda3e68c29ad8e0da2d48ff0e5c0be94ebfbdd2d2e1d45d963a0d813420
IUniswapV2Factory.sol	2a1a7de8091a4c3b92e6aee83c721106e3d80a99c4552d79d7a51c04cc1aa519
IUniswapV2Pair.sol	37395c7354bca9a7a282d30dbf0928e5f62b2db4045a3d93dabccf3efa5a6e2d
IUniswapV2Router01.sol	15d50e9129927b7f130b2218650789fd69111c1f52ed40c219dddf21fd4560f2
IUniswapV2Router02.sol	f2ba10aec9911956ceedc672dfeb9968624dd8b9c320ceeae0cbbb5bc92dd03b
SafeMath.sol	15941f3904992a62ed117e93d9e2d5c4c22bd09a7ff97fdd5f49273cf09703ac

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	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
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `swappingOnlyFromContract` to true and adding the owner's address to the `_AddressesClearedForSwap` mapping.

```
if (swappingOnlyFromContract) {
    if (automatedMarketMakerPairs[from]) {
        require(_AddressesClearedForSwap[recipient], "You are not allowed to
SWAP directly on Pancake");
    }
    if (automatedMarketMakerPairs[recipient]) {
        require(_AddressesClearedForSwap[from], "You are not allowed to SWAP
directly on Pancake");
    }
}
```

Additionally, the contract owner can abuse the tokenTransferFeeDistributor address in order to produce unexpected values. As a result, the distributeTokenTransferFee can either revert or manipulate the transactions.

```
if (!swapping) {
    if (from != address(0) && (recipient == uniswapV2Pair || recipient ==
priceStabilizer) && !feeExempt[from]) { // sell token
        swapping = true;
        amount -= distributeTokenTransferFee(from, amount);
        swapping = false;
```

In the same manner, the contract owner can abuse the priceStabilizer address in order to produce unexpected values.

```
if (!priceStabilizingDisabled) {
    if (from != address(0) && recipient == uniswapV2Pair) { // sell token
        if (!swapping) {
            swapping = true;
            require(priceStabilizer != address(0), "null price stabilizer");
            amount = IPriceRegulator(priceStabilizer).regulateSell(from,
amount);
            swapping = false;
        }
```

The contract owner has also the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `antiWhaleAmount` to the maximum allowed value. As a result, the antiWhaleAmount will always be greater than the amount.

```
require(
    antiWhaleAmount == 0 ||
    amount <= antiWhaleAmount ||
        !automatedMarketMakerPairs[from] ||
        recipient == address(priceStabilizer),
    "You are not permitted to swap buy more than #antiWhale tokens"
);
```

Recommendation

The contract could embody a check for not allowing setting the antiWhaleAmount 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.

Regarding the external addresses read more in the corresponding section.

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L163

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `blacklistAddress` function.

```
require(
    !_blackListedAddresses[from] && !_blackListedAddresses[recipient],
    "Wallet is blacklisted"
);
```

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. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	ZD	Zero Division
	US	Untrusted Source
	CR	Code Repetition
	MC	Missing Check
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L06	Missing Events Access Control
	L07	Missing Events Arithmetic

ZD - Zero Division

Criticality	minor
Location	contract.sol#L457,475

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert.

```
IFeeDistributor _feeDistributor = IFeeDistributor(tokenTransferFeeDistributor);
uint256 denominator = _feeDistributor.getMaxFeeResolution();
uint256 sent = 0;
uint256 i;
uint256 feeCount = _feeDistributor.getFeeCount();

for (i = 0; i < feeCount; i ++) {
    uint256 feeAmount = _totalAmount * _feeDistributor.getFeeShare(i) /
denominator;
    transferFrom(from, _feeDistributor.getFeeAddress(i), feeAmount);
    sent += feeAmount;
}
```

Recommendation

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

US - Untrusted Source

Criticality	critical
Location	contract.sol

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result it may produce security issues and harm the transactions.

Many features of the contract are working as a delegator to other contracts. For instance, the auto generated liquidity pool and the swap features are delegated to a contract called "priceStabilizer". The secured functionality of these contracts is essential for the Deck contract.

The auditing of these contracts are out of the audit scope.

```
IFeeDistributor _feeDistributor = IFeeDistributor(tokenTransferFeeDistributor);
//
amount = IPriceRegulator(priceStabilizer).regulateSell(from, amount);
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

MC - Missing Check

Criticality	critical
Location	contract.sol

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The properties nodeCreationFeeRewardsPoolShare and nodeCreationFeeLiquidityPoolShare could revert the transaction if the value is more than 10000, since the expression `amountToBeDistributed - rewardsPoolTokens - swapTokens` will underflow.

```
uint256 rewardsPoolTokens = amountToBeDistributed
    .mul(nodeCreationFeeRewardsPoolShare)
    .div(10000);

if (rewardsPoolTokens > 0) {
    super._transfer(address(this), rewardPool, rewardsPoolTokens);
}

// Liquidity pool gets half Deck token and half BUSD
uint256 swapTokens = amountToBeDistributed
    .mul(nodeCreationFeeLiquidityPoolShare)
    .div(10000);
_swapAndLiquify(swapTokens);

distributeBUSD(amountToBeDistributed - rewardsPoolTokens - swapTokens);
```

The property feeOnNodeCashout could revert the transaction if the value is more than 10000, since the expression `rewardAmount - feeAmount` will underflow.

```
uint256 feeAmount = 0;
if (feeOnNodeCashout > 0) {
    feeAmount = rewardAmount.mul(feeOnNodeCashout).div(10000);
    super._transfer(rewardPool, address(this), feeAmount);
```

```
    distributeBUSD(feeAmount);
}

return rewardAmount - feeAmount;
```

Recommendation

The sum of nodeCreationFeeRewardsPoolShare, nodeCreationFeeLiquidityPoolShare and feeOnNodeCashout should be less than 10000. The sum of nodeCreationFeeRewardsPoolShare and nodeCreationFeeLiquidityPoolShare should also be less than 10000.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract/Deck.sol#L100,146,259,354,419

Description

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

```
setAntiWhale
distributeNodeCreationFees
setAutomatedMarketMakerPair
updateUniswapV2Router
initialize
```

Recommendation

Use the external attribute for functions never called from the contract.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract/Deck.sol#L92,96,105,423,428,433,437,441,447,465,33

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.

```
_AddressesClearedForSwap
_totalAmount
_set
_user
_feeDistributor
_priceStabilizer
_enable
_tokenForSwapPair
_tokenToPayWithAddr
...
```

Recommendation

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

L06 - Missing Events Access Control

Criticality	minor
Location	contract/Deck.sol#L92

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.

```
nodeManagerAddr = _nodeManagerAddr
```

Recommendation

Emit an event for critical parameter changes.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract/Deck.sol#L100,237,245,419

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.

```
antiWhaleAmount = amount
nodeCreationFeeRewardsPoolShare = rewardFee
distributionThreshold = newVal
distributionThreshold = swapAmount
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Deck	Implementation	ERC20Upgradeable, GovernableUpgradeable		
	setSwappingOnlyFromContract	External	✓	onlyGovernor
	allowSwap	Internal	✓	
	disallowSwap	Internal	✓	
	blackListAddr	External	✓	onlyGovernor
	removeFromBlackList	External	✓	onlyGovernor
	isBlackListed	External		-
	setNodeManagerAddr	External	✓	onlyGovernor
	setSwapPairToken	Public	✓	onlyGovernor
	initialize	Public	✓	initializer
	updateUniswapV2Router	Public	✓	onlyGovernor
	_beforeTokenTransfer	Internal	✓	
	transferFrom	Public	✓	-
	setdistributionThreshold	External	✓	onlyGovernor
	setRewardPoolAddr	External	✓	onlyGovernor
	setFees	External	✓	onlyGovernor
	setEnableTrading	External	✓	onlyGovernor
	setAutomatedMarketMakerPair	Public	✓	onlyGovernor
	_setAutomatedMarketMakerPair	Private	✓	
	_swapAndLiquify	Private	✓	
	swapBusdForDeck	External	✓	-
	swapDeckForBusd	External	✓	-
	_swapTokensForBUSD	Private	✓	
	payForNode	External	✓	onlyNodeManager
	distributeNodeCreationFees	Public	✓	onlyGovernor

	_distributeNodeCreationFees	Private	✓	
	_feeOnCashout	Private	✓	
	cashoutRewardToNoder	External	✓	onlyNodeManager
	setAntiWhale	Public	✓	onlyGovernor
	enablePriceStabilizing	External	✓	onlyGovernor
	setPriceStabilizer	External	✓	onlyGovernor
	updateTokenTransferFeeDistributor	External	✓	onlyGovernor
	updateBUSDDistributor	External	✓	onlyGovernor
	setFeeExempt	External	✓	onlyGovernor
	distributeTokenTransferFee	Internal	✓	
	distributeBUSD	Internal	✓	
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		
	__ERC20_init	Internal	✓	onlyInitializing
	__ERC20_init_unchained	Internal	✓	onlyInitializing
	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	✓	

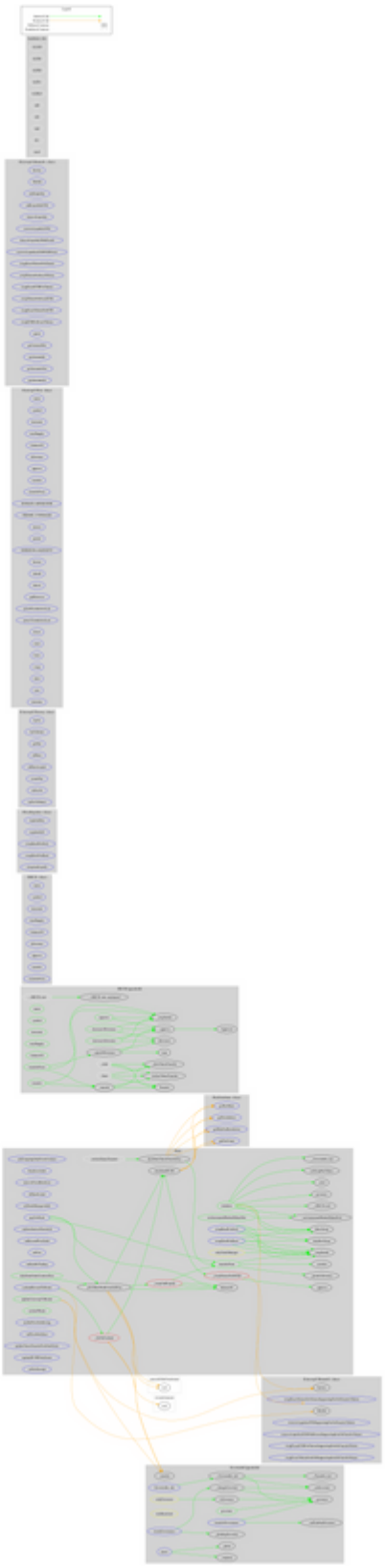
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
GovernableUpgradeable	Implementation	Initializable, PausableUpgradeable		
	Governable_init	External	✓	initializer
	__Governable_init	Internal	✓	onlyInitializing
	governor	Public		-
	_governor	Internal		
	_pendingGovernor	Internal		
	isGovernor	Public		-
	_setGovernor	Internal	✓	
	_setPendingGovernor	Internal	✓	
	transferGovernance	External	✓	onlyGovernor
	claimGovernance	External	✓	-
	_changeGovernor	Internal	✓	
	pause	External	✓	onlyGovernor
IERC20	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
IFeeDistributor	Interface			
	getMaxFeeResolution	External	✓	-
	getFeeCount	External	✓	-
	getFeeShare	External	✓	-
	getFeeAddress	External	✓	-

IPriceRegulator	Interface			
	regulateBuy	External	✓	-
	regulateSell	External	✓	-
	swapBusdForDeck	External	✓	-
	swapDeckForBusd	External	✓	-
	swapAndLiquify	External	✓	-
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		-
	mint	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	✓	-
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		

Contract Flow



Domain Info

Domain Name	web.app
Registry Domain ID	300A2C851-APP
Creation Date	2019-01-08T22:05:04Z
Updated Date	2021-12-12T09:32:53Z
Registry Expiry Date	2023-01-08T22:05:04Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	http://www.markmonitor.com
Registrar	MarkMonitor Inc.
Registrar IANA ID	292

The domain has been created in 6 months before the creation of the audit.

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, blacklisting addresses and external source manipulation. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. 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

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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io