CheckPoint

Token Security Audit Report
Prepared for LinkSync

[v.1.0]

November 2021

Document Properties

Client	LinkSync
Platform	Binance Smart Chain
Language	Solidity
Codebase	0xF6a22B0593df74F218027A2d8b7953c9b4542AA1

Audit Summary

Delivery Date	04.11.2021
Audit Methodology	Static Analysis, Manual Review
Auditor(s)	Erno Patiala
Classification	Publlic
Version	1.0

Contact Information

Company	CheckPoint
Name	Hanna Järvinen
Telegram	t.me/checkpointreport
E-mail	contact@checkpoint.report

Remark: For more information about this document and its contents, please contact CheckPoint team

Table Of Contents

1 Executive Summary

On 04/11/2021, CheckPoint conducted a full audit for the LinkSync to verify the overall security posture including a smart contract review to discover issues and vulnerabilities in the source code. Static Code Analysis, Dynamic Analysis, and Manual Review werdone in conjunction to identify smart contract vulnerabilities together with technical & business logic flaws that may be exposed to the potential risk of the platform and the ecosystem.

After further analysis and internal discussion, we determined a few issues of varying severities that need to be brought up and paid more attention to. More information can be found in **Section 5 'Audit Result'**. Practical recommendations are provided according to each vulnerability found and should be followed to remediate the issue.

LinkSync (SYNC)
Low Risk Level

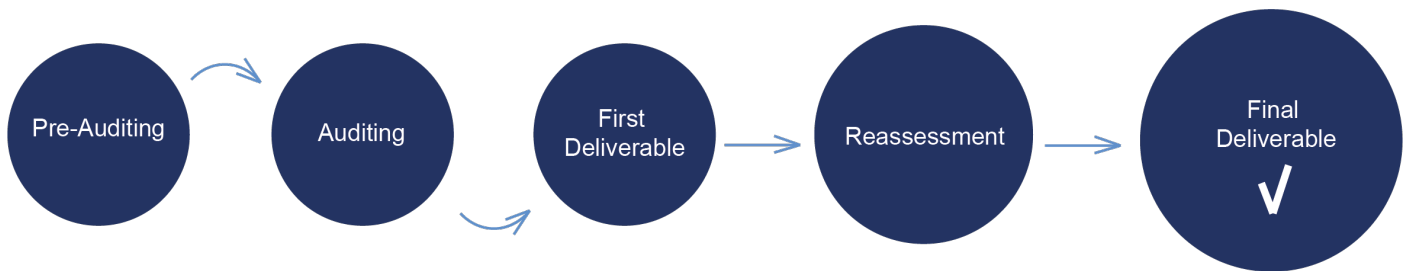|	
Communication Channels	Website Content Analysis, Social Media Listening
Smart Contract Code	Smart Contract Details, Contract Function Details, Issues Checking Status, Detailed Findings Information

THIS TOKEN PASSES CHECKPOINT'S SECURITY VERIFICATION STANDART

2 Audit Methodology

CheckPoint conducts the following procedure to enhance the security level of our clients' tokens:

- **Pre-Auditing**

  Planning a comprehensive survey of the token, its ecosystem, possible risks & prospects, getting to understand the overall operations of the related smart contracts, checking for readiness, and preparing for the auditing.

- **Auditing**

  Study of all available information about the token on the Web, inspecting the smart contracts using automated analysis tools and manual analysis by a team of professionals.

- **First Deliverable and Consulting**

  Delivering a preliminary report on the findings with suggestions on how to remediate those issues and providing consultation.

- **Reassessment**

  Verifying the status of the issues and whether there are any other complications in the fixes applied.

- **Final Deliverable**

  Providing a full report with the detailed status of each issue.

The security audit process of CheckPoint includes three types testing:

    1.   Examining publicly available information about the token on social networks, including a detailed overview of the official website and analysis of the latest messages and opinions about the token.

    2.   Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.

    3.   Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.
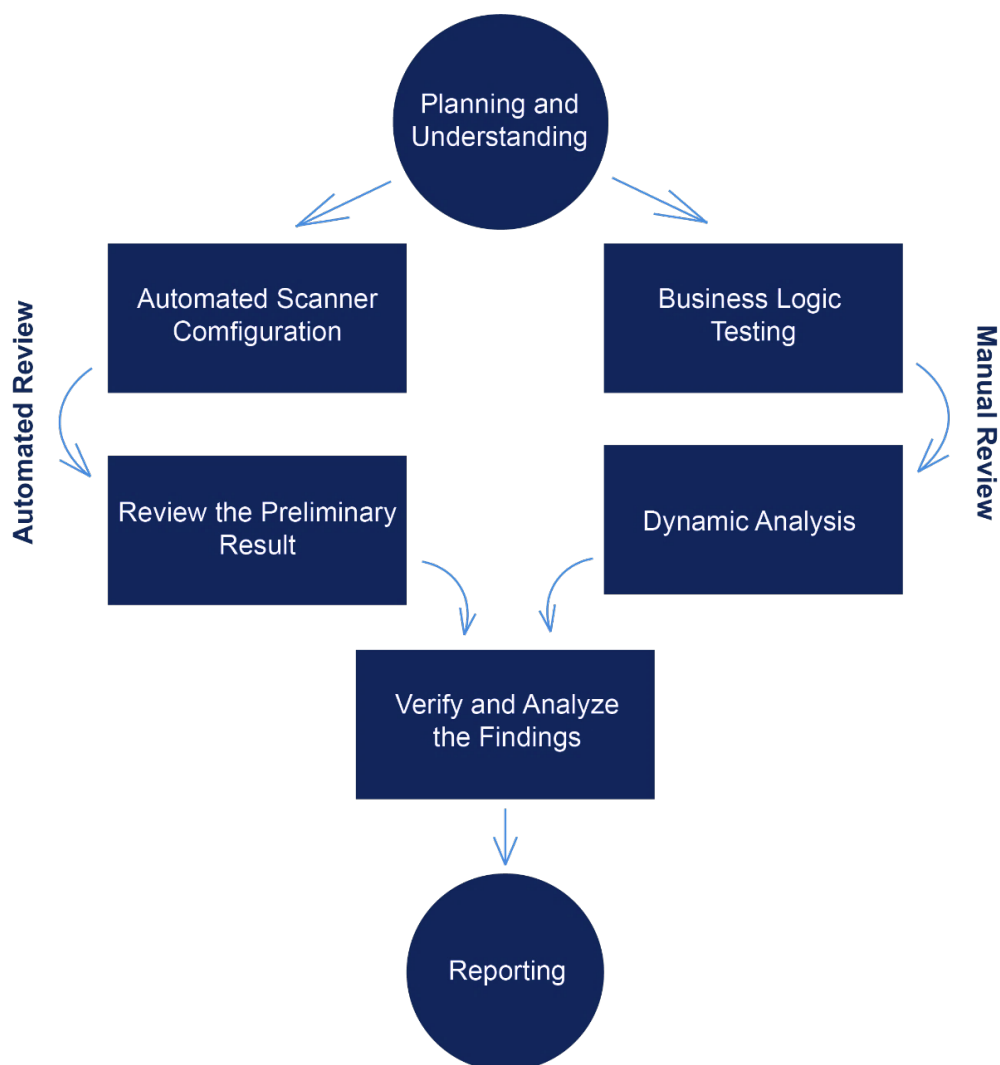
Remark: Manual and Automated review approaches can be mixed and matched including business logic analysis in terms of malicious doers' perspective

In particular, we perform the audit according to the following procedure:

- **Planning & Understanding**
    - determine scope of testing and understand application purpose and workflows;
    - identify key risk areas, including technical and business risks;
    - determine approach – which sections to review within the resource constraints and review method – automated, manual or mixed.

- **Automated Review**
    - adjust automated source code review tools to inspect the code for known unsafe coding patterns;
    - verify output of the tool in order to eliminate false positive result, and if necessary, adjust and re-run the code review tool.

- **Manual Review**
    - testing for business logic flaws requires thinking in unconventional methods;
    - identify unsafe coding behavior via static code analysis.

- **Reporting**
    - analyze the root cause of the flaws;
    - recommend coding process improvements.

3 Risk Level Classification

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology:

- **Likelihood** represents how likely a particular vulnerability is to be uncovered and exploited in the wild.

- **Impact** measures the technical loss and business damage of a successful attack.

- **Severity** demonstrates the overall criticality of the risk and calculated as the product of impact and likelihood values, illustrated in a twodimensional matrix. The shading of the matrix visualizes the different risk levels.

	Low	**Medium**	**High**
Low	Weakness	Low	Medium
Medium	Low	Medium	High
High	Medium	High	Critical

IMPACT (vertical axis)

LIKELIHOOD (horizontal axis)

Remark: Likelihood and Impact are categorized into three levels: H, M, and L, i.e., High, Medium and Low respectively. Severity is determined by likelihood and impact and can be classified into five categories accordingly, i.e., Critical, High, Medium, Low and Weakness

For prioritization of the vulnerabilities, we have adopted the scheme by five distinct levels for risk: Critical, High, Medium, Low, and Weakness. The risk level definitions are presented in table.

LEVEL	DESCRIPTION
Weakness	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering
Low	Low severity vulnerabilities may affect the operation of the DeFi project in certain scenarios
Medium	Medium severity vulnerability will affect the operation of the DeFi project. It is recommended to fix medium-risk vulnerabilities
High	High severity vulnerabilities will affect the normal operation of the DeFi project. It is strongly recommended to fix high-risk vulnerabilities
Critical	Critical severity vulnerabilities will have a significant impact on the security of the DeFi project

4 Project Overview

4.1 Communication Channels

https://linksync.tech/
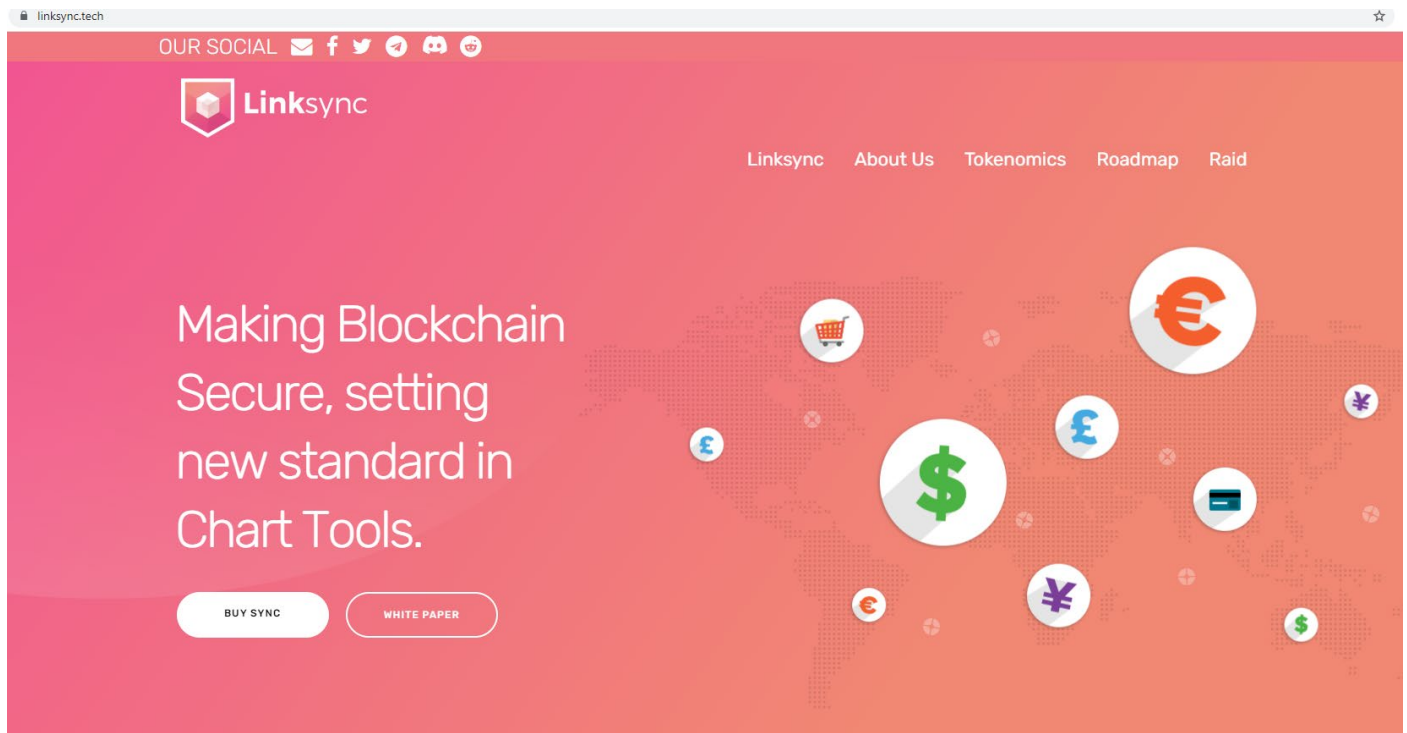
Above the image is an actual snapshot of the current live website of the project.

✓ Mobile Friendly	✓ 5 Social Media Networks
✓ No JavaScript Errors	✓ 2500+ Telegram Members
✓ Visionary Roadmap	✓ 7000+ Twitter Followers
✓ Spell Check	✓ Active Voice Chats
✓ Valid SSL Certificate	✓ No Injected Spam and Popus Found

Remark: This page contains active links

4.2 Smart Contract Details

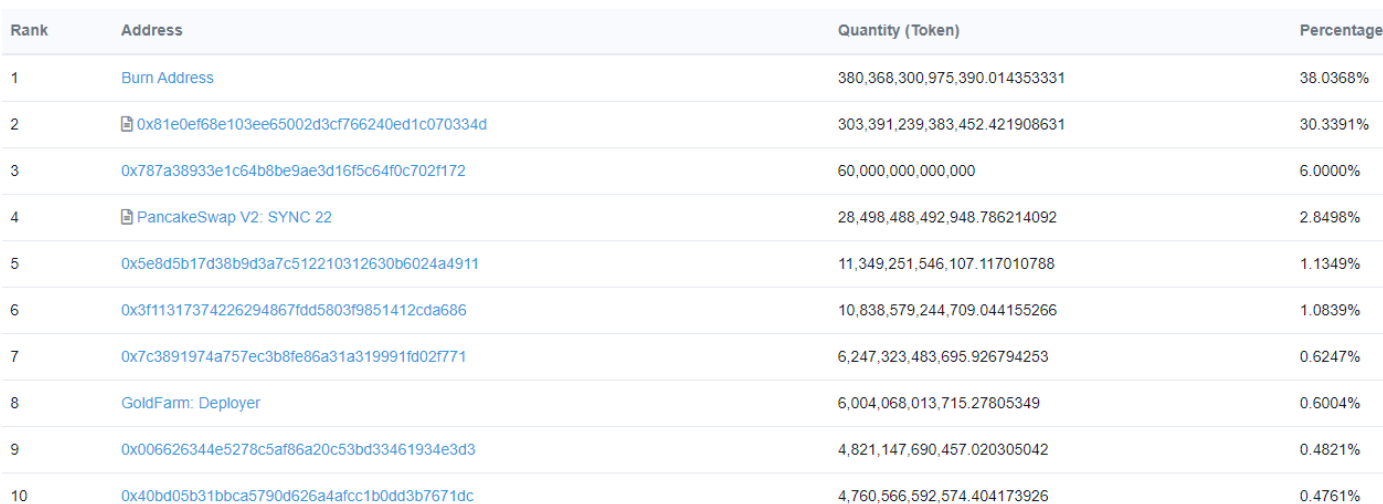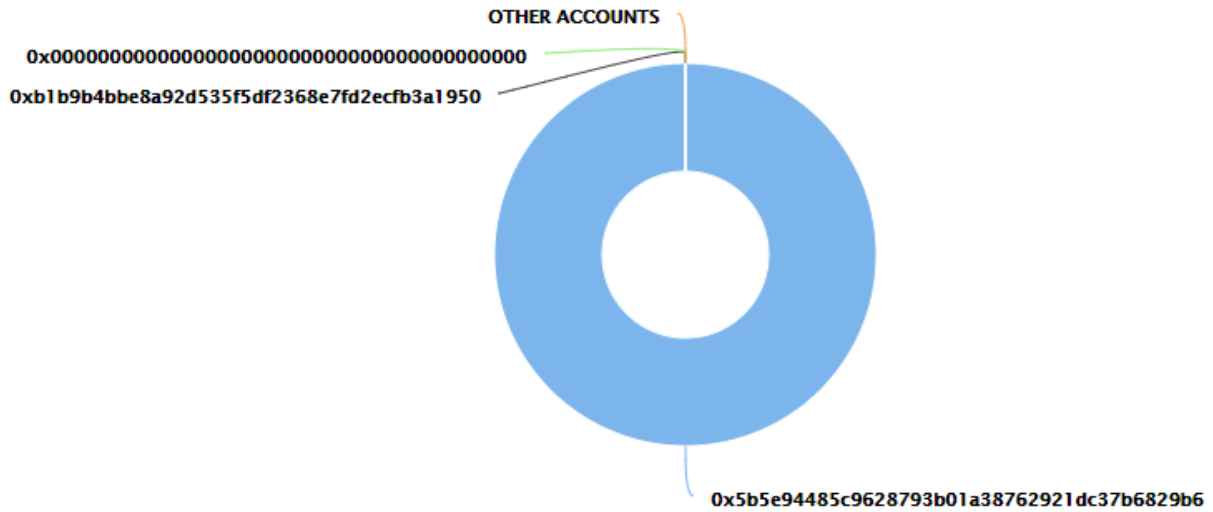|	
Contract Name	LinkSync
Contract Address	0xF6a22B0593df74F218027A2d8b7953c9b4542AA1
Total Supply	1,000,000,000,000,000
Token Ticker	SYNC
Decimals	9
Token Holders	1,608
Transactions Count	4,435
Top 10 Holders Dominance	81,63%
Liquidity Fee	8%
Tax Fee	4%
Marketing Wallet Address	0xdc8c43defae6859801b8dff25499f0f5810d41a4
Uniswap V2 Pair Contract	0x1c06a11e94b5502d011bbd240f23d1c147561dab
Contract Deployer Address	0x7Aa350bEA485A4EA8860AAA42F6D4aFb5b6df2fa
Current Owner Address	0xf35a242220d897f6031bae553256de754364974b

LinkSync Top 10 Token Holders

LinkSync Top 10 Token Holders

Source: BscScan.com

Rank	Address	Quantity (Token)	Percentage
1	Burn Address	380,368,300,975,390.014353331	38.0368%
2	0x81e0ef68e103ee65002d3cf766240ed1c070334d	303,391,239,383,452.421908631	30.3391%
3	0x787a38933e1c64b8be9ae3d16f5c64f0c702f172	60,000,000,000,000	6.0000%
4	PancakeSwap V2: SYNC 22	28,498,488,492,948.786214092	2.8498%
5	0x5e8d5b17d38b9d3a7c512210312630b6024a4911	11,349,251,546,107.117010788	1.1349%
6	0x3f11317374226294867fdd5803f9851412cda686	10,838,579,244,709.044155266	1.0839%
7	0x7c3891974a757ec3b8fe86a31a319991fd02f771	6,247,323,483,695.926794253	0.6247%
8	GoldFarm: Deployer	6,004,068,013,715.27805349	0.6004%
9	0x006626344e5278c5af86a20c53bd33461934e3d3	4,821,147,690,457.020305042	0.4821%
10	0x40bd05b31bbca5790d626a4afcc1b0dd3b7671dc	4,760,566,592,574.404173926	0.4761%

✓ **~38% tokens are permanently removed from circulation**

✓ **PancakeSwap holds ~3% of the token's supply as liquidity**

LinkSync Top 3 LP Token Holders

OTHER ACCOUNTS

0x0000000000000000000000000000000000000000

0xb1b9b4bbe8a92d535f5df2368e7fd2ecfb3a1950

0x5b5e94485c9628793b01a38762921dc37b6829b6

Rank	Address	Quantity (Token)	Percentage
1	0x5b5e94485c9628793b01a38762921dc37b6829b6	2,213.594362117865531399	99.9626%
2	0xb1b9b4bbe8a92d535f5df2368e7fd2ecfb3a1950	0.827098145864885629	0.0374%
3	0x0000000000000000000000000000000000000000	0.000000000000001	0.0000%

[RISK] 1 wallet have ~100% LP tokens

LinkSync Contract Interaction Details

● Token Transfers -●- Token Contracts Count -+- Outbound Transfers -+- Inbound Transfers -+- Unique Address Sent -+- Unique Address Received
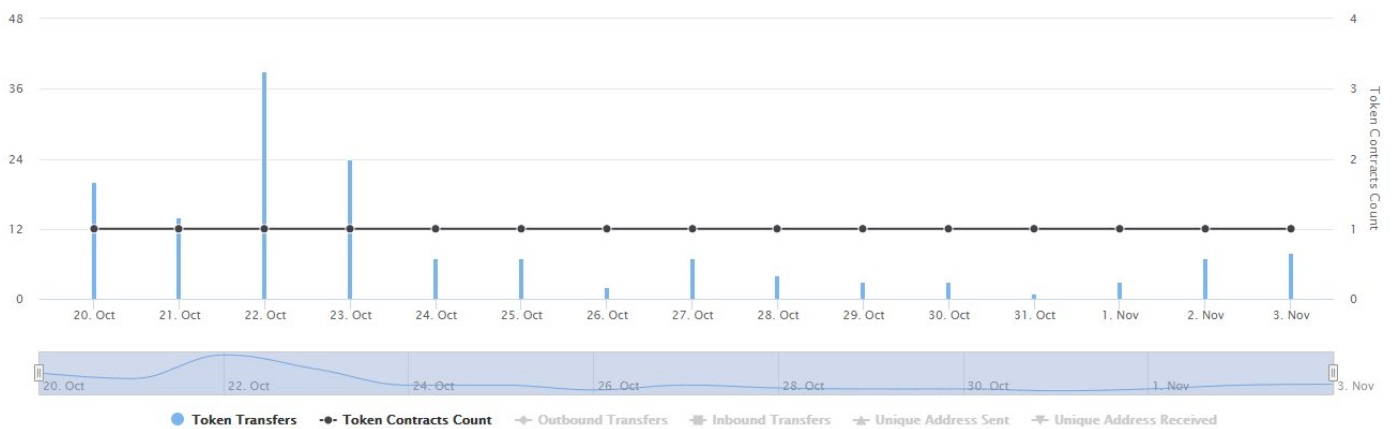
4.3 Contract Function Details

$ = payable function
# = non-constant function
[Int] = Internal
[Pub] = Public
[Prv] = Private
[Ext] = External

+ Context
   - [Int] _msgSender
   - [Int] _msgData

+ [Int] IERC20
   - [Ext] totalSupply
   - [Ext] balanceOf
   - [Ext] transfer #
   - [Ext] allowance
   - [Ext] approve #
   - [Ext] transferFrom #

+ [Lib] SafeMath
   - [Int] add
   - [Int] sub
   - [Int] sub
   - [Int] mul
   - [Int] div
   - [Int] div
   - [Int] mod
   - [Int] mod

+ [Lib] Address
   - [Int] isContract
   - [Int] sendValue #
   - [Int] functionCall #
   - [Int] functionCall #
   - [Int] functionCallWithValue #
   - [Int] functionCallWithValue #
   - [Prv] functionCallWithValue

+ Ownable (Context)
   - [Int] <Constructor> #
   - [Pub] owner
   - [Pub] renounceOwnership #
      - modifiers: onlyOwner
   - [Pub] transferOwnership #
      - modifiers: onlyOwner
   - [Pub] getUnlockTime
   - [Pub] getTime

       - modifiers: block.timestamp
   - [Pub] lock #
       - modifiers: onlyOwner

+ [Int] IUniswapV2Factory
   - [Ext] feeTo
   - [Ext] feeToSetter
   - [Ext] getPair
   - [Ext] allPairs
   - [Ext] allPairsLength
   - [Ext] createPair #
   - [Ext] setFeeTo #
   - [Ext] setFeeToSetter #

+ [Int] IUniswapV2Pair
   - [Ext] name
   - [Ext] symbol
   - [Ext] decimals
   - [Ext] totalSupply
   - [Ext] balanceOf
   - [Ext] allowance
   - [Ext] approve #
   - [Ext] transfer #
   - [Ext] transferFrom #
   - [Ext] DOMAIN_SEPARATOR
   - [Ext] PERMIT_TYPEHASH
   - [Ext] nonces
   - [Ext] permit #
   - [Ext] MINIMUM_LIQUIDITY
   - [Ext] factory
   - [Ext] token0
   - [Ext] token1
   - [Ext] getReserves
   - [Ext] price0CumulativeLast
   - [Ext] price1CumulativeLast
   - [Ext] kLast
   - [Ext] mint #
   - [Ext] burn #
   - [Ext] swap #
   - [Ext] skim #
   - [Ext] sync #
   - [Ext] initialize #

+ [Int] IUniswapV2Router01
   - [Ext] factory
   - [Ext] WETH
   - [Ext] addLiquidity #
   - [Ext] addLiquidityETH $
   - [Ext] removeLiquidity #
   - [Ext] removeLiquidityETH #

- [Ext] removeLiquidityWithPermit #
- [Ext] removeLiquidityETHWithPermit #
- [Ext] swapExactTokensForTokens #
- [Ext] swapTokensForExactTokens #
- [Ext] swapExactETHForTokens $
- [Ext] swapTokensForExactETH #
- [Ext] swapExactTokensForETH #
- [Ext] swapETHForExactTokens $
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
   - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
   - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
   - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
   - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens $
   - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ LinkSync (Context, IERC20, Ownable)
   - [Pub] <Constructor> #
   - [Pub] name
   - [Pub] symbol
   - [Pub] decimals
   - [Pub] totalSupply
   - [Pub] balanceOf
   - [Pub] transfer #
   - [Pub] allowance
   - [Pub] approve #
   - [Pub] transferFrom #
   - [Pub] increaseAllowance #
   - [Pub] decreaseAllowance #
   - [Pub] isExcludedFromReward
   - [Pub] totalFees
   - [Pub] minimumTokensBeforeSwapAmount
   - [Pub] buyBackUpperLimitAmount
   - [Pub] deliver #
   - [Pub] reflectionFromToken
   - [Pub] tokenFromReflection
   - [Pub] excludeFromReward #
      - modifiers: onlyOwner
   - [Ext] includeInReward #
      - modifiers: onlyOwner
   - [Prv] _approve #
   - [Prv] _transfer #
   - [Prv] swapTokens #
   - [Prv] buybackTokens #
   - [Prv] swapTokensForEth #

- [Prv] swapETHForTokens #
- [Prv] addLiquidity #
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] calculateTaxFee
- [Prv] calculateLiquidityFee
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee
  - modifiers: onlyOwner
- [Ext] setTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setLiquidityFeePercent #
  - modifiers: onlyOwner
- [Ext] setMaxTxPercent #
  - modifiers: onlyOwner
- [Ext] setMarketingDivisor #
  - modifiers: onlyOwner
- [Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
- [Ext] setBuybackUpperLimit #
  - modifiers: onlyOwner
- [Ext] setMarketingAddress #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Pub] setBuyBackEnabled #
  - modifiers: onlyOwner
- [Prv] transferToAddressETH #

4.4 Issues Checking Status

CHECKING ITEM	NOTES	RESULT
Arbitrary Jump with Function Type Variable	N / A	PASS
Arithmetic Accuracy Deviation	N / A	PASS
Assert Violation	N / A	PASS
Authorization through tx.origin	N / A	PASS
Business Logic	N / A	PASS
Code with No Effects	N / A	PASS
Critical Solidity Compiler	N / A	PASS
Delegatecall to Untrusted Callee	N / A	PASS
Design Logic	N / A	PASS
DoS with Block Gas Limit	N / A	PASS
DoS with Failed Call	N / A	PASS
Function Default Visibility	N / A	PASS
Hash Collisions With MVLA	N / A	PASS
Incorrect Constructor Name	N / A	PASS
Incorrect Inheritance Order	N / A	PASS
Integer Overflows and Underflows	N / A	PASS
Lack of Proper Signature Verification	N / A	PASS
Message Call with Hardcoded Gas Amount	N / A	PASS
Missing Protection Against SRA	N / A	PASS
Presence of Unused Variables	N / A	PASS
Reentrancy	N / A	PASS
Requirement Violation	N / A	PASS

CHECKING ITEM	NOTES	RESULT
Right-To-Left-Override Control Character	N / A	PASS
Shadowing State Variables	N / A	PASS
Signature Malleability	N / A	PASS
State Variable Default Visibility	N / A	PASS
Timestamp Dependence	N / A	PASS
Transaction Order Dependence	N / A	PASS
Typographical Error	N / A	PASS
Unencrypted Private Data On-Chain	N / A	PASS
Unexpected Ether balance	N / A	PASS
Uninitialized Storage Pointer	N / A	PASS
Use of Deprecated Solidity Functions	N / A	PASS
Weak Sources of Randomness From CA	N / A	PASS
Write to Arbitrary Storage Location	N / A	PASS

Remark: To evaluate the risk, we go through a list of check items and each would be labeled with a severity category. For one check item, if our tool or analysis does not identify any issue, the contract is considered safe regarding the check item

4.5 Detailed Findings Information

[RISK] DoS with Block Gas Limit

- The function _getCurrentSupply uses the loop for evaluating total supply and total reward. It could be aborted with out-of-gas exception if there will be a long excluded addresses list. Including an account in the reward again may result in unexpected behavior.

```solidity
function _getCurrentSupply() private view returns(uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation: Consider removing the _getCurrentSupply function. If this is not desired, consider avoiding it, especially on accounts with a significant balance.

- The function includeInReward uses the loop to find and remove addresses from the _excluded list. It could be aborted with out-of-gas exception if there will be a long excluded addresses list. Including an account in the reward again may result in unexpected behavior.

```solidity
ftrace | funcSig
function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

Recommendation: Consider removing the includeInReward function. If this is not desired, consider avoiding it, especially on accounts with a significant balance.

[RISK] Owner Privileges (in the period when the owner is not renounced)

The contract contains the following privileged functions that are restricted by the onlyOwner.

- The owner of the contract can lock and unlock the contract.

```
function lock(uint256 time↑) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time↑;
    emit OwnershipTransferred(_owner, address(0));
}
```

```
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(block.timestamp > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

- The owner of the contract can exclude and include accounts from fees and reward distribution.

```
function excludeFromFee(address account↑) public onlyOwner {
    _isExcludedFromFee[account↑] = true;
}
```

```
function includeInFee(address account↑) public onlyOwner {
    _isExcludedFromFee[account↑] = false;
}
```

```
function excludeFromReward(address account↑) public onlyOwner() {
    // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router.');
    require(!_isExcluded[account↑], "Account is already excluded");
    if(_rOwned[account↑] > 0) {
        _tOwned[account↑] = tokenFromReflection(_rOwned[account↑]);
    }
    _isExcluded[account↑] = true;
    _excluded.push(account↑);
}

ftrace | funcSig
function includeInReward(address account↑) external onlyOwner() {
    require(_isExcluded[account↑], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account↑) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account↑] = 0;
            _isExcluded[account↑] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The owner of the contract can change a marketing wallet address.

```
ftrace | funcSig
function setMarketingAddress(address _marketingAddress↑) external onlyOwner() {
    marketingAddress = payable(_marketingAddress↑);
}
```

- The owner can set a 'liquidity fee' and a 'tax fee'.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner() {
    taxFee = taxFee↑;
}

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner() {
    liquidityFee = liquidityFee↑;
}
```

- The owner can set a max tax percentage.

```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount↑) external onlyOwner() {
    maxTxAmount = maxTxAmount↑;
}
```

- The owner can set swap and liquify enable status.

```
function setSwapAndLiquifyEnabled(bool _enabled↑) public onlyOwner {
    swapAndLiquifyEnabled = _enabled↑;
    emit SwapAndLiquifyEnabledUpdated(_enabled↑);
}
```

5 Audit Result

L E V E L

I S S U E S

Weakness
Low

DoS with Block Gas Limit (2)

Owner Privilegies (6)

1. The contract utilizes SafeMath libraries along with following the ERC20 standard.

2. The owner of contract can exclude and include accounts from transfer fees and reward distribution at any time.

3. There is a 'Liquidity fee' and a 'Tax Fee' on all transactions for any non-excluded address that participates in a transfer.

4. The owner of contract can to modify these fees to any percentage at any time. This might deter investors as they could be wary that these fees might one day be set to 100% to force transfers to go to the contract owner.

5. There is a transfer limit on the number of tokens which can be sent in a single transaciton which can be updated by the owner of the contract. The owner can also lock tokens in any address.

6. The owner of contract has the ability to use the lock function in order to set ownership to address(0) at any time. This can be used in a nefarious way by the project team to restore ownership and change fee structures.

5.1 Findings Summary

LinkSync
Low Risk Level

- ✓ **No external vulnerabilities were identified within the smart contract's code**

- ✓ **The code is fully customized**

- ✓ **As with any presale, ensure trust in the team prior to investing**

- ✓ **Ensure trust in the team as they have substantial control over the ecosystem and will control the charity/marketing wallets**

- ✓ **LinkSync token was audited, and no issues were found**

6 Disclamer

CheckPoint team issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these. For the facts that occurred or existed after the issuance, CheckPoint is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to CheckPoint by the information provider till the date of the insurance report. CheckPoint is not responsible for the background and other conditions of the project.

This security audit is not produced to supplant any other type of assessment and does not guarantee the discovery of all security vulnerabilities within the scope of the assessment. However, we warrant that this audit is conducted with goodwill, professional approach, and competence. Since an assessment from one single party cannot be confirmed to cover all possible issues within the smart contract(s), CheckPoint suggests conducting multiple independent assessments to minimize the risks. Lastly, nothing contained in this audit report should be considered as investment advice.