



MOTECH AUDIT

SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

1INCH PROGRESS REPORT

2021

12 OCT 2021

SECURITY ASSESMENT

TABLE OF CONTENTS

Summary

Background

Audit Details

Disclaimer

Contract Details

Token Distribution

Contract Interaction Details

Top 10 Token Holders



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

Finding

BAH-01 : User-Defined Getters
BAH-02 : SafeMath Redundancy
CON-01 : Whitelist Evaluation
GFR-01 : Misleading Error Message
GFR-02 : Inexistent Access Control
GMH-01 : Incorrect Stake Notification Implementation
LVH-01 : `SafeMath` Redundancy
MFG-01 : `SafeMath` Redundancy
MFG-02 : Variable Typos
MFG-03 : `require` to `modifier`
MGH-01 : Incorrect `to` Conditional
MGH-02 : Redundant `isDefault` Invocation
MOO-01 : Event Indexing
MOO-02 : Disproportionate Initial Minting
MOO-03 : Incorrect Transfer Evaluation
MOO-04 : Midway Condition Evaluation
MOO-05 : Loop Optimization
MOO-06 : Function Visibility Optimization
MOO-07 : Incorrect Implementation / Naming Convention
MOO-08 : `require` Consistency
MOO-09 : Inexistent Input Sanitization
MOO-10 : Documentation Consistency
MOO-11 : Redundant Use of Dynamic Arrays
REW-01 : `SafeMath` Redundancy
RFR-01 : `SafeMath` Redundancy
RFR-02 : Unfair Proportionate Calculation
RFR-03 : Prohibition of Ether Transfers
RFR-04 : Inexistent Input Sanitization of Mooniswap Addresses
SQR-01 : Babylonian Method Optimization
UER-01 : SafeMath Redundancy
UER-02 : Incorrect len Boundry Check
UER-03 : Assembly-based Optimization
UER-04 : Redundant Function Implementation
UER-05 : Potential Code Redundancy
VBH-01 : `SafeMath` Redundancy
VOT-01 : Redundant Nesting

Conclusion

SECURITY ASSESSMENT

SUMMARY

This report has been prepared for 1inch Mooniswap v2 to discover issues and vulnerabilities in the sourcecode of the 1inch Mooniswap v2 project as well as any contract dependencies that were not part of anofficially recognized library. A comprehensive examination has been performed, utilizing Static Analysis andManual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts producedby industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommendaddressing these findings to ensure a high level of security standards and industry practices. We suggestrecommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified inpublic;
- Provide more transparency on privileged activities once the protocol is live.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESMENT

BACKGROUND

MotechAudit was commissioned by 1INCH Token to perform an audit of smart contracts:

<https://etherscan.io/address/0x11111111117dc0aa78b770fa6a738034120c302>

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

AUDIT DETAILS



AUDITED PROJECT

1INCH



DEPLOYER ADDRESS

0x514DF5293Aa7cA53C05D79C37b836596C4ABf687



CLIENT CONTACTS:

1INCH Token team



BLOCKCHAIN

ETHEREUM Project



WEBSITE:

<https://1inch.io/>



SECURITY ASSESSMENT

DISCLAIMER

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and MotechAudit and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (MotechAudit) owe no duty of care towards you or any other person, nor does MotechAudit make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and MotechAudit hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, MotechAudit hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against MotechAudit, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

CONTRACT DETAILS

Token contract details for Dec-24-2020

Contract name	1inch
Contract address	0x111111111117dc0aa78b770fa6a738034120c302
Total supply	1,500,000,000
Token ticker	1INCH Token (1INCH)
Decimals	18
Token holders	73,241
Transactions count	955,893
Top 100 holders dominance	97.11%
Contract deployer address	0x514DF5293Aa7cA53C05D79C37b836596C4ABf687
Contract's current owner address	0x514DF5293Aa7cA53C05D79C37b836596C4ABf687

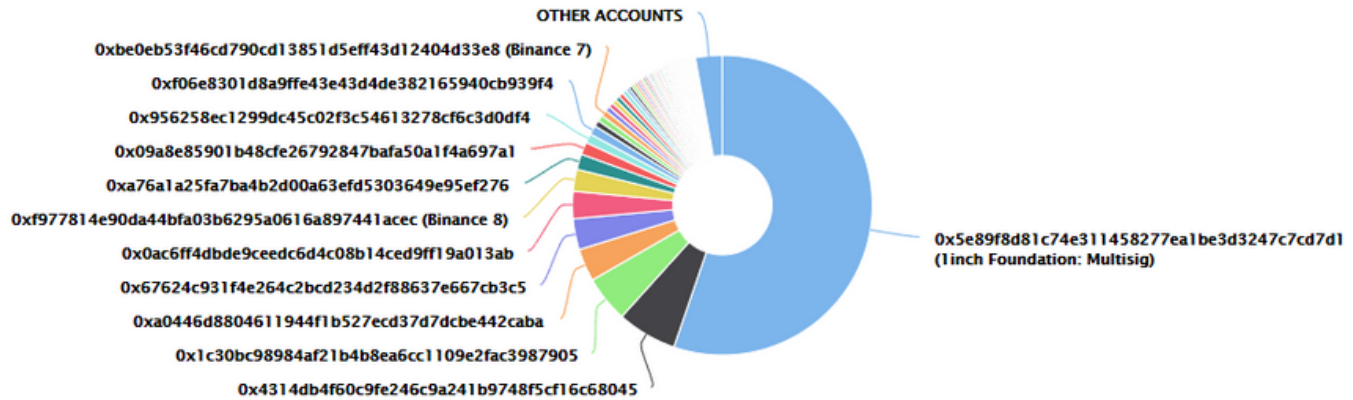


SECURITY ASSESMENT

IINCH TOKEN DISTRIBUTION

IINCH Token Top 100 Token Holders

Source: Etherscan.io



(A total of 1,456,645,194.50 tokens held by the top 100 accounts from the total supply of 1,500,000,000.00 token)

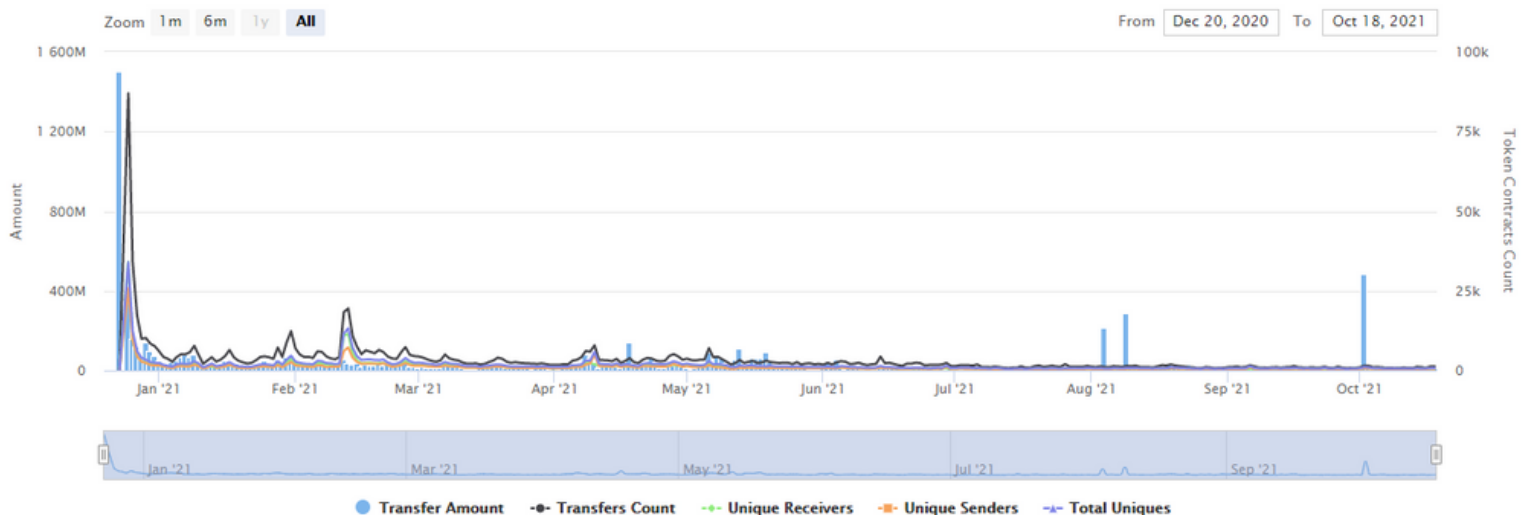
IINCH TOKEN CONTRACT INTERACTION DETAILS

Time Series: Token Contract Overview

Wed 23, Dec 2020 - Mon 18, Oct 2021

Token Contract 0x11111111117dc0aa78b770fa6a738034120c302 (IINCH Token)




















Source: Etherscan.io



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

TOP 10 TOKEN HOLDERS

Rank	Address	Quantity	Percentage	Value	Analytics
1	 1inch Foundation: Multisig	828,932,718.494516890874324592	55.2622%	\$2,917,843,169.10	
2	 0x4314db4f60c9fe246c9a241b9748f5cf16c68045	96,500,000	6.4333%	\$339,680,000.00	
3	 0x1c30bc98984af21b4b8ea6cc1109e2fac3987905	75,000,000	5.0000%	\$264,000,000.00	
4	 0xa0446d8804611944f1b527ecd37d7dcbe442caba	52,896,644.888017357776074438	3.5264%	\$186,196,190.01	
5	 0x67624c931f4e264c2bcd234d2f88637e667cb3c5	50,000,000	3.3333%	\$176,000,000.00	
6	 0x0ac6ff4dbde9ceedc6d4c08b14ced9ff19a013ab	44,295,000	2.9530%	\$155,918,400.00	
7	Binance 8	35,332,879.1537354251502	2.3555%	\$124,371,734.62	
8	 0xa76a1a25fa7ba4b2d00a63efd5303649e95ef276	25,000,000	1.6667%	\$88,000,000.00	
9	 0x09a8e85901b48cfe26792847bafa50a1f4a697a1	20,000,000	1.3333%	\$70,400,000.00	
10	 0x956258ec1299dc45c02f3c54613278cf6c3d0df4	15,000,000	1.0000%	\$52,800,000.00	

source:etherscan.io



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESMENT

FINDINGS



Critical	0 (0.00%)
Major	0 (0.00%)
Medium	3 (8.33%)
Minor	5 (13.89%)
Informational	28 (77.78%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
BAH-01	User-Defined Getters	Gas Optimization	Informational	Resolved
BAH-02	SafeMath Redundancy	Gas Optimization	Informational	Resolved
CON-01	Whitelist Evaluation	Control Flow	Minor	Resolved
GFR-01	Misleading Error Message	Inconsistency	Informational	Resolved
GFR-02	Inexistent Access Control	Control Flow	Minor	Resolved
GMH-01	Incorrect Stake Notification Implementation	Logical Issue	Medium	Resolved
LVH-01	SafeMath Redundancy	Gas Optimization	Informational	Resolved
MFG-01	SafeMath Redundancy	Gas Optimization	Informational	Resolved



ID	Title	Category	Severity	Status
MFG-02	Variable Typos	Coding Style	● Informational	👍 Resolved
MFG-03	require to modifier	Gas Optimization	● Informational	🕒 Partially Resolved
MGH-01	Incorrect to Conditional	Logical Issue	● Minor	👍 Resolved
MGH-02	Redundant isDefault Invocation	Gas Optimization	● Informational	👍 Resolved
MOO-01	Event Indexing	Inconsistency	● Informational	👍 Resolved
MOO-02	Disproportionate Initial Minting	Logical Issue	● Minor	👍 Resolved
MOO-03	Incorrect Transfer Evaluation	Logical Issue	● Minor	👍 Resolved
MOO-04	Midway Condition Evaluation	Gas Optimization	● Informational	🕒 Partially Resolved
MOO-05	Loop Optimization	Gas Optimization	● Informational	🕒 Partially Resolved
MOO-06	Function Visibility Optimization	Gas Optimization	● Informational	🕒 Partially Resolved
MOO-07	Incorrect Implementation / Naming Convention	Gas Optimization	● Informational	👍 Resolved
MOO-08	require Consistency	Inconsistency	● Informational	👍 Resolved
MOO-09	Inexistent Input Sanitization	Logical Issue	● Medium	🕒 Pending
MOO-10	Documentation Consistency	Inconsistency	● Informational	👍 Resolved
MOO-11	Redundant Use of Dynamic Arrays	Inconsistency	● Informational	👍 Resolved
REW-01	SafeMath Redundancy	Gas Optimization	● Informational	👍 Resolved



ID	Title	Category	Severity	Status
RFR-01	SafeMath Redundancy	Gas Optimization	● Informational	✓ Resolved
RFR-02	Unfair Proportionate Calculation	Mathematical Operations	● Informational	✓ Resolved
RFR-03	Prohibition of Ether Transfers	Logical Issue	● Informational	✓ Resolved
RFR-04	Inexistent Input Sanitization of Mooniswap Addresses	Logical Issue	● Medium	✓ Resolved
SQR-01	Babylonian Method Optimization	Gas Optimization	● Informational	⌚ Partially Resolved
UER-01	SafeMath Redundancy	Gas Optimization	● Informational	✓ Resolved
UER-02	Incorrect len Boundry Check	Logical Issue	● Informational	✓ Resolved
UER-03	Assembly-based Optimization	Gas Optimization	● Informational	⌚ Partially Resolved
UER-04	Redundant Function Implementation	Gas Optimization	● Informational	⌚ Partially Resolved
UER-05	Potential Code Redundancy	Gas Optimization	● Informational	✓ Resolved
VBH-01	SafeMath Redundancy	Gas Optimization	● Informational	✓ Resolved
VOT-01	Redundant Nesting	Language Specific	● Informational	⌚ Partially Resolved



SECURITY ASSESSMENT

BAH-01 | USER-DEFINED GETTERS

Category	Severity	Location	Status
Gas Optimization	● Informational	utils/BalanceAccounting.sol: 11, 14~16	🟢 Resolved

Description

The linked variables contain user-defined getter functions that are equivalent to their name barring for an underscore (_) prefix / suffix.

Recommendation

We advise that the linked variables are instead declared as public and that they are renamed to their respective getter's name as compiler-generated getter functions are less prone to error and much more maintainable than manually written ones

Alleviation

The 1inch team decided to leave the variable as is to properly depict that it is meant to be altered only by the internal functions of the BalanceAccounting contract.



SECURITY ASSESSMENT

BAH-02 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	utils/BalanceAccounting.sol: 29	✓ Resolved

Description

The linked statement conducts a SafeMath subtraction operation on the `_totalSupply` variable based on the amount burned from the `_balances` mapping. As the contract level variables of the contract are private, they cannot be altered by derivative implementations and as such, the `_totalSupply` is kept in perfect sync with the amount of total balance held in the `_balances` mapping. As a result, if a SafeMath subtraction operation succeeds on the `_balances` mapping it is guaranteed to succeed in the `_totalSupply` variable as well, rendering its use redundant gaswise.

Recommendation

We advise that the SafeMath utilization from this point is omitted. For the sake of clarity, a comment maybe added to aid in future auditing endeavors.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability.



SECURITY ASSESSMENT

CON-01 | WHITELIST EVALUATION

Category	Severity	Location	Status
Control Flow	● Minor	utils/Converter.sol: 100	✓ Resolved

Description

The error message alludes that ETH transfers are completely forbidden to the contract whereas transfers from contracts are allowed.

Recommendation

We advise that the error message is revised to properly reflect the check's purpose.

Alleviation

The issue was acknowledged but no action was taken to alter the error message as it is believed to be sufficiently descriptive. To note, the contracts have been revised in the latest commit hash to instead rely on a receive implementation in the Converter contract.

SECURITY ASSESSMENT

GFR-02 | INEXISTENT ACCESS CONTROL

Category	Severity	Location	Status
Control Flow	● Minor	governance/GovernanceFeeReceiver.sol: 23~25, 27~31	✓ Resolved

Description

The linked functions are meant to conduct sensitive operations on the contract and utilize its full balance to conduct a swap. Even though the target of a given swap operation will always be the rewards address specified during construction time, a malicious path can lead to multiple pairs being exchanged artificially increasing volume and diminishing the final output sent to the rewards address.

Recommendation

We advise that both the unwrapLPTokens and swap functions are guarded via a check to ensure an authorized member is conducting those operations either via a governance or ownership system.

Alleviation

The 1inch team reported that sufficient ACL checks are imposed in the form of valid conversion checks within the Converter.sol implementation whereby a path of fixed length and whitelisted addresses is guaranteed, a slippage check is imposed as well as a check that at most 1% of available liquidity within the pool is swapped. These are considered sufficient measures against malicious trades.



SECURITY ASSESMENT

GMH-01 | INCORRECT STAKE NOTIFICATION IMPLEMENTATION

Category	Severity	Location	Status
Logical Issue	● Medium	inch/GovernanceMothership.sol: 49	✓ Resolved

Description

The notify For function is meant to be used to allow any address to force a status update of another address arbitrarily to ensure the stakes states are kept in sync. However, the second argument passed to the _notify For function is the balance of the msg.sender instead of the account to-be-notified, causing an incorrect balance to be reported for it and potentially for as many accounts as a particular msg.sender wants.

Recommendation

We advise that the second argument is instead changed to the balance Of of the actual account in the same way the batch Notify For implementation functions.

Alleviation

The balance Of measurement was properly fixed in this commit hash as per our recommendation

SECURITY ASSESSMENT

LVH-01 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/LiquidVoting.sol: 30~31, 34	✓ Resolved

Description

The linked mathematical operations utilize their SafeMath counterpart implementation whereas it is completely redundant for the operations of L31 and L34 as well as potentially redundant for the operation of L30.

Recommendation

We advise that its usage is omitted for the guaranteed operations and evaluated for the potential operation to optimize gas costs.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability

SECURITY ASSESSMENT

MFG-01 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	governance/MooniswapFactoryGovernance.sol: 147, 149	🟢 Resolved

Description

The linked statements both contain redundant operations as they are nested within an if-else block that guarantees their validity

Recommendation

We advise that these SafeMath statements are omitted.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability

SECURITY ASSESMENT

MFG-02 | VARIABLE TYPOS

Category	Severity	Location	Status
Coding Style	● Informational	governance/MooniswapFactoryGovernance.sol: 162~163, 166, 167	🟢 Resolved

Description

The linked code lines contain a misspelling of the default word as defaul , leading to mistyped variablenames.

Recommendation

We advise that these variable names are corrected

Alleviation

The variable names were corrected according to our recommendation in the linked commit hash

SECURITY ASSESSMENT

MFG-03 | REQUIRE TO MODIFIER

Category	Severity	Location	Status
Gas Optimization	● Informational	governance/MooniswapFactoryGovernance.sol: 98, 107, 116, 117, 126~127, 136	🔄 Partially Resolved

Description

The linked require statements could instead be coded in a small internal or private function that is invoked by a corresponding modifier to reduce the gas footprint of the contract.

Recommendation

We advise that the suggestion explained in the description is implemented.

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SECURITY ASSESMENT

MGH-01 | INCORRECT TO CONDITIONAL

Category	Severity	Location	Status
Logical Issue	● Minor	governance/MooniswapGovernance.sol: 102	✓ Resolved

Description

The linked `_before` Token Transfer hook incorrectly sanitizes burn operations by evaluating whether the `from` instead of the `to` variable is different than the zero address for the ternary operator used on the `balanceTo` assignment.

Recommendation

We advise that the ternary evaluation instead utilizes the `to` variable

Alleviation

The ternary operator was properly fixed according to our recommendation in the linked commit hash.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESMENT

MGH-02 | REDUNDANT ISDEFAULT INVOCATION

Category	Severity	Location	Status
Gas Optimization	● Informational	governance/MooniswapGovernance.sol: 142, 143	✓ Resolved

Description

The if conditional that precedes those two statements already evaluates whether the vote is the default one, meaning a literal can be passed here instead.

Recommendation

We advise that a literal is instead passed here to optimize the gas cost of those statements.

Alleviation

The optimization we advised was applied in this commit hash.



SECURITY ASSESMENT

MOO-01 | EVENT INDEXING

Category	Severity	Location	Status
Inconsistency	● Informational	Mooniswap.sol: 57	✓ Resolved

Description

The Swapped event is indexing the source token, however, the destination token remains unindexed

Recommendation

We advise that either both or neither of the two tokens involved in the swap are indexed to ensure consistency in the codebase.

Alleviation

After conversing with the 1inch team, we identified that our initial suggestion was incorrect as it is not possible to add more than 3 indexed fields to an event and as such, this exhibit is considered void

SECURITY ASSESSMENT

MOO-02 | DISPROPORTIONATE INITIAL MINTING

Category	Severity	Location	Status
Logical Issue	● Minor	Mooniswap.sol: 140~153	✓ Resolved

Description

The initial minting process of a Mooniswap appears to be disproportionate in the sense that it does not rely on the initial deposit, meaning that a user can simply deposit only a single unit and acquire a mint equal to `_BASE_SUPPLY.mul(99)` as only a `Math.max` operation is used on L145.

Recommendation

We advise that the validity of such a mint is evaluated and the corresponding code segments are refactored if it is deemed unfair.

Alleviation

The 1inch team responded by stating that the initial mint donates 1% of the minted amount to the pool to ensure that underfunded pools cannot be redeemed i.e. a deposit of a single unit for the creation of the pool will render that unit unredeemable. Additionally, consequent mints are properly evaluated to be proportionate thus diminishing the percentage of the original mint

SECURITY ASSESMENT

MOO-03 | INCORRECT TRANSFER EVALUATION

Category	Severity	Location	Status
Logical Issue	● Minor	Mooniswap.sol: 150~151	✓ Resolved

Description

The linked code invokes uniTransfer From for an amount equal to maxAmounts[i] and consequently assigns that value to received Amounts[i] even if a token imposes fees on the transaction.

Recommendation

We advise that the uniBalance Of evaluation paradigm that is already utilized in L174 is also utilized here to ensure compatibility with tokens that carry a transfer fee

Alleviation

The 1inch team stated that the actual received amount can be safely ignored for the first deposit as it is not weighted for the minting process 1inch



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

MOO-04 | MIDWAY CONDITION EVALUATION

Category	Severity	Location	Status
Gas Optimization	● Informational	Mooniswap.sol: 147, 169	⌚ Partially Resolved

Description

The linked require statements ensure that the maxAmounts provided to the contract at the start are greater-than (>) zero, yet they are evaluated mid-way through.

Recommendation

We advise that both elements of the maxAmounts array are evaluated at the very start to ensure no gas is wasted in the statements that precede them if the function is going to fail regardless

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SECURITY ASSESMENT

MOO-05 | LOOP OPTIMIZATION

Category	Severity	Location	Status
Gas Optimization	● Informational	Mooniswap.sol: 155~156	🕒 Partially Resolved

Description

The linked code block contains four for loops to ultimately conduct all operations necessary on the 2-member maxAmounts and realBalances arrays.

Recommendation

We advise that the block is refactored to utilize a single loop as we believe some or all of the for loops can be grouped into one and significantly reduce gas cost.

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SECURITY ASSESMENT

MOO-06 | FUNCTION VISIBILITY OPTIMIZATION

Category	Severity	Location	Status
Gas Optimization	● Informational	Mooniswap.sol: 130, 134, 191, 195	🕒 Partially Resolved

Description

The linked functions are declared as external or public, contain array function arguments and are meant to be mostly invoked by external parties.

Recommendation

We advise that the functions' visibility specifiers are set to public or external and the array-based arguments change their data location from memory to calldata, optimizing the gas costs of the functions

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

MOO-07 | INCORRECT IMPLEMENTATION / NAMING CONVENTION

Category	Severity	Location	Status
Gas Optimization	● Informational	Mooniswap.sol: 322~327	🟢 Resolved

Description

The naming convention utilizing tax alludes to a tax being imposed on the balances, however L323 assigns the addition instead of the subtraction of the tax from the srcBalance.

Recommendation

We advise that either the naming convention or the implementation are adjusted to properly reflect what they are meant to achieve.

Alleviation

The 1inch team revised the codebase in the linked commit hash to contain more legible variable naming.

SECURITY ASSESSMENT

MOO-08 | REQUIRE CONSISTENCY

Category	Severity	Location	Status
Inconsistency	● Informational	Mooniswap.sol: 209	🟢 Resolved

Description

The project imposes require checks on multiple functions, like L255 of `_doTransfers`, to ensure that zero-value swaps etc. are prohibited. The `withdrawFor` function fails to impose such a check as it merely ensures that the value is greater-than-or-equal to `minReturns[i]` which itself could be 0.

Recommendation

We advise that an additional check akin to `_doTransfers` is imposed here as well to ensure consistency.

Alleviation

After consulting with the 1inch team, we concluded that the require check imposed in the linked lines are actually carried out properly as tokens with a small number of decimal places can indeed yield a zero amount whereas their counterpart within the pool may yield a non-zero amount. Consequently, this exhibit is nullified.

SECURITY ASSESSMENT

MOO-09 | INEXISTENT INPUT SANITIZATION

Category	Severity	Location	Status
Logical Issue	● Medium	Mooniswap.sol: 222	ⓘ Pending

Description

The src and dst arguments of the swapFor function are not guaranteed to be equal to the supported tokens of the exchange. While _doTransfers will simply yield 0 for the confirmed variable on unsupported tokens, this can lead to incorrect reward minting on _mintRewards as the confirmed balance is simply utilized during reward calculation

Recommendation

We advise that the if conditional of _doTransfers is instead changed to a require check to ensure only the supported tokens are ever swapped on the Mooniswap exchange.

Alleviation

The 1inch team stated that a check is actually imposed within _doTransfers on the result of the invocation of _getReturn which will yield 0 for unsupported tokens, thus preventing unsupported tokens from being accepted by the _doTransfers function.



SECURITY ASSESSMENT

MOO-10 | DOCUMENTATION CONSISTENCY

Category	Severity	Location	Status
Inconsistency	● Informational	Mooniswap.sol: 304~316	✓ Resolved

Description

The linked documentation block is meant to explain the calculations carried out in `_getReturn` , however the variable naming conventions of the function block do not conform to what is laid out in the documentation and the documentation itself appears inconsistent with the statements of the function.

Recommendation

We advise that either the documentation or the variable naming conventions are updated correspondingly to increase the legibility of this particular code block.

Alleviation

The 1inch team stated that the accompanying comments are meant to paint the greater picture with regards to the formula the Mooniswap system is utilizing whereas the last segment, $(ret = dx * y / (x + dx) * (x + dx - slip_fee * dx) / (x + dx))$, is what is being implemented by the formula of the function.



SECURITY ASSESSMENT

MOO-11 | REDUNDANT USE OF DYNAMIC ARRAYS

Category	Severity	Location	Status
Inconsistency	● Informational	Mooniswap.sol: 191, 195	🟢 Resolved

Description

The functions of the Mooniswap contract accept statically sized arrays of 2 members whereas the withdraw prefixed functions accept dynamically sized arrays whose size is ensured to be 2 within their respective code block.

Recommendation

We advise that they too are adjusted to be statically sized arrays to reduce the gas cost of invoking the functions and ensuring consistency in the codebase.

Alleviation

The 1inch team stated that the arrays are dynamic to allow users to optionally specify minReturns. However, for the sake of gas optimization, we still believe the system can be revised as a value of 0 on both members would render them optional since the require conditional value $\geq \text{minReturns}[i]$ would always yield true due to the usage of uint256.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

REW-01 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	governance/rewards/Rewards.sol: 75, 77, 97, 99, 101	✓ Resolved

Description

The linked statements all conduct SafeMath operations whilst the safety of their operations is guaranteed by the system either via if-else clauses or constant variable utilizations

Recommendation

We advise that the SafeMath utilizations from these points are omitted. For the sake of clarity, comments may be added to aid in future auditing endeavors.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability.

SECURITY ASSESSMENT

RFR-01 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	ReferralFeeReceiver.sol: 57, 72	✓ Resolved

Description

The linked statement conducts a SafeMath subtraction operation on the `_totalSupply` variable based on the amount burned from the `_balances` mapping. As the contract level variables of the contract are private, they cannot be altered by derivative implementations and as such, the `_totalSupply` is kept in perfect sync with the amount of total balance held in the `_balances` mapping. As a result, if a SafeMath subtraction operation succeeds on the `_balances` mapping it is guaranteed to succeed in the `_totalSupply` variable as well, rendering its use redundant gaswise.

Recommendation

We advise that the SafeMath utilization from this point is omitted. For the sake of clarity, a comment maybe added to aid in future auditing endeavors.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability

SECURITY ASSESSMENT

RFR-02 | UNFAIR PROPORTIONATE CALCULATION

Category	Severity	Location	Status
Mathematical Operations	● Informational	ReferralFeeReceiver.sol: 185~195	🟢 Resolved

Description

The linked code block is meant to collect an epoch's share by calculating the percentage of shares a user has proportionate to the total supply of shares of a given epoch and multiply the resulting percentage with the total inch balance of the epoch.

Recommendation

The calculation carried out is unfair because the share is subtracted from the total supply after it has been claimed, resulting in disproportionate percentages due to rounding down of the calculation on L190. For a minimal example: State A: Share Supply: 100 1inch Balance: 50 User A: 20 shares equivalent to $20 * 50 / 100 \rightarrow 10$ User B: 5 shares equivalent to $5 * 50 / 100 \rightarrow 2.5 \rightarrow 2$ User B claims State B: Share Supply: 95 1inch Balance: 48 User A: 20 shares equivalent to $20 * 48 / 95 \rightarrow 10.105$ The differences should be negligible in most cases as they too are rounded down, however for larger amounts or after compounding they may become significant. We advise that the impact of this is evaluated and potentially ignored if identified to be completely negligible in the context of Mooniswap.

Alleviation

After discussing with the 1inch team, we concluded that this issue is negligible as the 1inch token has 18 decimal places.

SECURITY ASSESSMENT

RFR-03 | PROHIBITION OF ETHER TRANSFERS

Category	Severity	Location	Status
Logical Issue	● Informational	ReferralFeeReceiver.sol: 10	✓ Resolved

Description

The contract does not permit Ethereum transfers from contracts in contrast to GovernanceFeeReceiver

Recommendation

We advise that support for Ether transfers from contracts is added as is the case with GovernanceFeeReceiver.sol .

Alleviation

The codebase was revised to instead relocate the prevention of Ether transfers from GovernanceFeeReceiver to Converter thus avoiding code duplication and ensuring consistency



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

RFR-04 | INEXISTENT INPUT SANITIZATION OF MOONISWAP ADDRESSES

Category	Severity	Location	Status
Logical Issue	● Medium	ReferralFeeReceiver.sol: 38, 52, 66, 113, 129, 141, 166, 174, 200	🟢 Resolved

Description

The linked code blocks link to function implementations within ReferralFeeReceiver that do not conduct any sanitization on the input mooniswap variable and invoke functions on it.

Recommendation

We advise that a sanitization check is imposed whereby the address is verified to be existent within the Mooniswap Factory mapping via its isPool exposed function.

Alleviation

The 1inch team implemented a validPool modifier within Converter to ensure that the pools are properly validated to have been deployed by the official Mooniswap factory. Additionally, the team added reentrancy guards to ensure that even in the event of valid pools malicious tokens do not attempt to re-enter the contract and attempt to exploit it.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESMENT

SQR-01 | BABYLONIAN METHOD OPTIMIZATION

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/Sqrt.sol: 8~22	🔒 Partially Resolved

Description

The linked Babylonian Square Root implementation can be further optimized by omitting the final elseblock as Solidity returns zeroed out values by default regardless of whether the return variable has been explicitly named or not. Additionally, we highly advise that the new, more optimized Babylonian Method from ABDK Consulting is evaluated as an optimized alternative which has already been integrated in Uniswap.

Recommendation

Included in the description.

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

UER-01 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/UniERC20.sol: 42	✓ Resolved

Description

The linked mathematical statement conducts a SafeMath operation whilst its safety is guaranteed by the preceding if clause.

Recommendation

We advise that the SafeMath usage is omitted from the codebase to optimize gas cost.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability.

SECURITY ASSESMENT

UER-02 | INCORRECT LEN BOUNDRY CHECK

Category	Severity	Location	Status
Logical Issue	● Informational	libraries/UniERC20.sol: 66	✓ Resolved

Description

The boundary check of the len variable is meant to ensure that the len variable can fit within a single 8-bit slot by ensuring its within its valid range. However, the upper bound specified is incorrect.

Recommendation

We advise that the upper bound is adjusted to a strict less-than (<) comparison with 256 as a byte cannot retain the number 256 since its valid range is between 0-255 in its unsigned representation.

Alleviation

The 1inch team responded by stating that the check is not meant to ensure the length fits within 1 bytebut rather that the length is simply within the specified bound as they decided to not support tokens with aname larger than 256 bytes



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

UER-03 | ASSEMBLY-BASED OPTIMIZATION

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/UniERC20.sol: 78~82	🔄 Partially Resolved

Description

The linked code block creates a new in-memory bytes array and assigns each byte sequentially to it via a loop from the data array

Recommendation

We advise that this code block is refactored in assembly to be heavily optimized. The EVM is meant to operate on 32-byte datasets and as such, the array could be copied 32-bytes at a time at a much cheaper gas cost than it currently is being done so.

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

UER-04 | REDUNDANT FUNCTION IMPLEMENTATION

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/UniERC20.sol: 86, 89~91	⌚ Partially Resolved

Description

The function `_toHex` is implemented twice utilizing function overloading, however the middleware implementation accepting an address argument is only used once.

Recommendation

We advise that the middleware implementation of `_toHex` is omitted and that L86 invokes the actual implementation directly to avoid the superfluous function call.

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SECURITY ASSESSMENT

UER-05 | POTENTIAL CODE REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/UniERC20.sol: 86	🟢 Resolved

Description

The linked line performs an on-chain conversion from the address low-level representation to its human-readable hexadecimal representation.

Recommendation

As the conversion between raw bytes and hexadecimal does not affect its usability as a unique key, we advise that its actual need is evaluated as the binary to hex conversion could potentially happen completely off-chain to optimize the gas costs necessitated by on-chain operations.

Alleviation

The 1inch team stated that the purpose of the on-chain conversion is to aid in the readability of tokennames in blockchain explorers like Etherscan, thus nullifying the validity of this exhibit.

SECURITY ASSESSMENT

VBH-01 | SAFEMATH REDUNDANCY

Category	Severity	Location	Status
Gas Optimization	● Informational	libraries/VirtualBalance.sol: 32, 33, 36	🟢 Resolved

Description

The linked statements conduct SafeMath operations when their safety may be guaranteed by the system for the first and last linked statements and is fully guaranteed for the middle statement.

Recommendation

We advise that the SafeMath utilizations are evaluated and omitted where possible to optimize gas costs.

Alleviation

The 1inch team preferred to leave the redundancy as is for the sake of readability.

SECURITY ASSESSMENT

VOT-01 | REDUNDANT NESTING

Category	Severity	Location	Status
Language Specific	● Informational	libraries/Vote.sol: 7~9	🔄 Partially Resolved

Description

The vote library is meant to be utilized on Data structs that contain only a single uint256 value.

Recommendation

We advise that the library is instead applied directly on the uint256 data type to decrease code ambiguity when considering nested statements such as L22-L24.

Alleviation

The 1inch Mooniswap v2 development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SECURITY ASSESMENT

CONCLUSION



Smart contracts contain owner privileges!

Motech Audit note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT