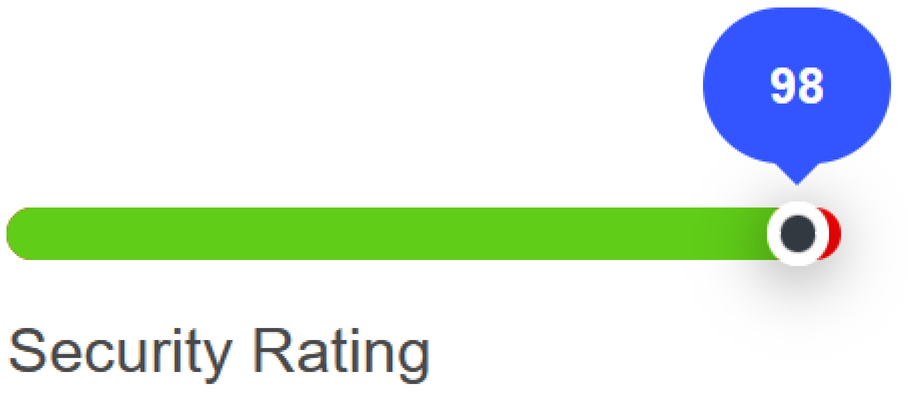
SECURICHAIN

SMART CONTRACT SECURITY ANALYSIS REPORT ON

MECH MASTER

Mar 3rd 2022



(The rating is based on the number, severity and latest status of detected issues)

Disclaimer

This report containing confidential information which can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

SecuriChain does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed.

The report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

TABLE OF CONTENTS

1. VULNERABILITY ASSESSMENT OVERVIEW

1.1. ASSIGNING RISK LEVELS

The Auditor categorizes each of the detected vulnerabilities into 4 levels (**High**, **Medium**, **Low**, and **Info**) according to the degree of the risks it may cause in Customer's operations. For details of the rating standards, please refer to "Appendix 2 Risk Rating." Please also note that the assessment of the findings is based on Auditor's own perspective and may contain speculations in some cases.

1.2. SCOPE OF WORK

Project Name	MECH MASTER
Platform	ETHEREUM
Languages	SOLIDITY
Methods	AUTOMATION SCAN, ARCHITECTURE REVIEW, FUNCTIONAL TESTING, MANUAL CODE REVIEW
Repository	**MARKETPLACE** COMMIT: *E8BC34E* **STAKING & TOKEN** COMMIT: *D458AB1*
Documents	
Timelines	*Feb 16th 2022 - Mar 3rd 2022*

1.3. CHECKSUM FILE

MARKETPLACE

STT	Hash	Name
1	415429eb6ca1c133d1d4b96d0bc8e428eb45c3bfdb7bbeabb43e3bd1e920a3e4	AcceptedToken.sol
2	0e4b0ad4d83b2605575db3f6944daa83cf5abb14bdebfd256258a0984582e374	ERC1155Marketplace.sol
3	abef2d6b12c9088a852a9f7e8552da4f4660b7157ffe1fb40a0c39b07e8ee34b	ERC1155Test.sol
4	a8342a7af9ff6cb785e217cffe792bdcc0b7ef406cbb06518e2b94d6984c067d	ERC721Test.sol
5	1e38d5b8b360250b25df4dd795cfd4a74a4502ce2ff2d999b4e017330a36f4e4	Marketplace.sol
6	f8039fa05f2d174d057fbfaf6a173e623cf971f0ecf8d9f094d8edb2396acef7	TokenTest.sol
7	9107461b53e25c5e25e166440722c3a2740df906243a715abc3c1cfe3a970d71	WETH.sol
8	5bf39bafd15c8ff6ef577d3cee12494a73036ce047f00d8237e9e299eaddb392	IMarketplace.sol
9	55267256fc784ccf4fbf4d50347f78f40a1bb67be95dce281f1dd09547fac294	IWETH.sol

STAKING

STT	Hash	Name
1	7152b22f3316441234d7854dd15320888ddc4bebc768c76535d5ac1aa694dcfc	MechaRace.sol
2	8cef8afe569647f5e66ad5902742e5ec79fde317c9853367eea52a20a0902562	StakingPool.sol
3	fe9acdffaa3c59eb62f70dfbe07e8b4bf03b9b282a23c742881cac1170771fca	ERC20Mock.sol
4	329f10635acc6c23fb952d186e51cdb98c886edf2928783ba4537a786d40c0c5	ERC721Mock.sol

TOKEN

STT	Hash	Name
1	26045dca676ea5333ab248759a2423ca96b4ddab31148f7766d092052310f72d	Equipment.sol
2	d9211569779aee2fcf2d904765cb308309773d2c5ba27ced8e76a18c0ca92555	ERC1155Upgradeable.sol
3	387270f59480bdcd8a0f04dbb1f5171174f484397a78dd86705820a1db95cf8c	MechaCloner.sol
4	544924c5ad14e46ce2f8c3af6850657a2f2e40662ea8d471bd7148e806305679	Mecha.sol
5	fe13d32093f12f981d30a834575eff0a51dc791a6af559b5f8c92056343eb333	Nitro.sol
6	734d3988a6cabac3e573dd591e9b2ac4826aea4e76d3d17bfbc2e8b254d4b221	Pilot.sol
7	521fbff18b4a489a6bd27c9caad1c29845581077d4d4d9c3e5ea54be68e46c09	Token.sol
8	845fceafdc844b99719ac8285d5b2a6400bfb888c31350e5240e14e15570c497	IEquipment.sol
9	de41dfd4d48add5f14dfc95300d9694d8415c626688038edd28588cd34b6b1b3	IMecha.sol
10	2402bcdb17784c398a9bb665df31035eb3618afe398e2d00d24c6e933df68bc2	IPilot.sol

1.4. ASSESSMENT RESULTS

According to the assessment, the Customer's smart contracts have the security rating of 98/100

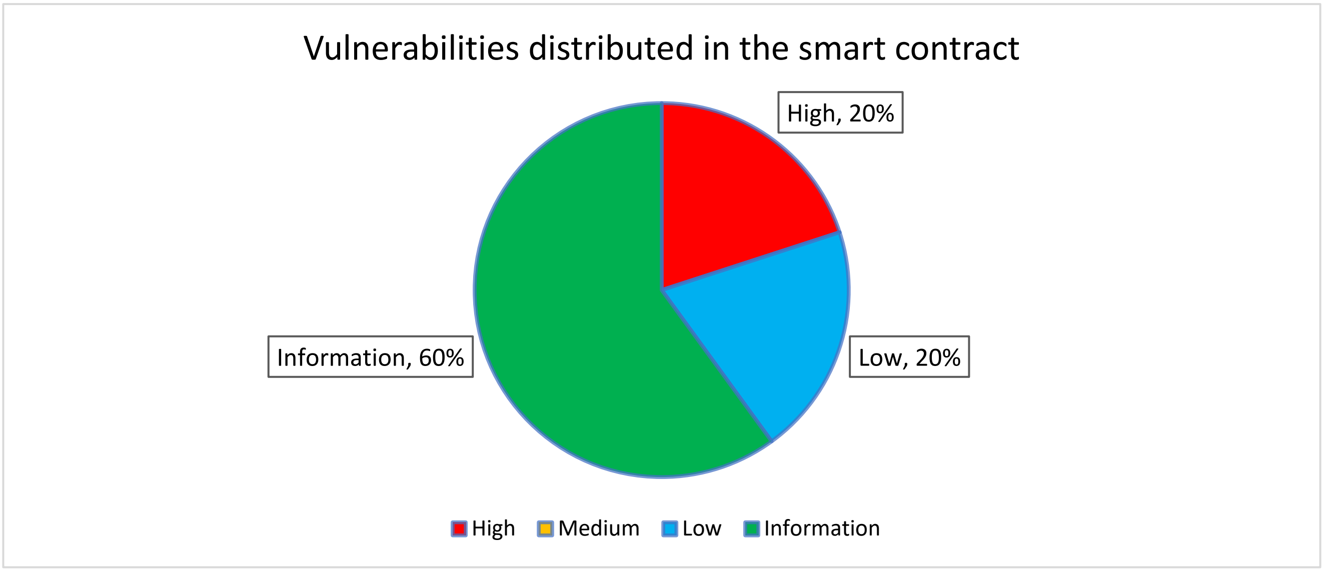
Rate	Description
96-100	**No vulnerabilities** were found or all detected ones have been resolved
70-95	Unresolved **Low-level** vulnerabilities exist
40-69	Unresolved **Medium-level** vulnerabilities exist
0-39	Unresolved **High-level** vulnerabilities exist

(*For more information on criteria for risk rating, refer to Appendix.2)*

2. FINDINGS

2.2. LIST OF VULNERABILITIES

The detected vulnerabilities are listed below. Please refer to "Appendix.2 Risk Rating" for the risk assessment method.



ID	Risk Level	Name	Amount	Status
SC1	Information	Unlocked Pragma	3	Unresolved
SC2	Low	Gas Optimization	1	Resolved in #b40af13 commit
SC3	High	Logic vulnerability	1	Resolved in #b40af13 commit

(For rating of each vulnerability, refer to Appendix 2.)

2.3. DETAILS

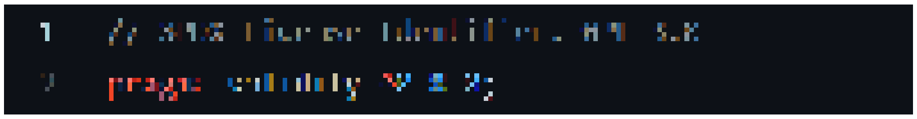
[1] Unlocked Pragma

3 INFO

- **Overview**

Contracts should be deployed with the same compiler version and flags that they have been thoroughly tested. Locking the pragma helps to ensure that contracts do not accidentally get deployed using.

- **Possible Impact**



(Blurring the image of the code snippet in the public report because the Customer's code is in the private repository)

An outdated compiler version that might introduce bugs that affect the contract system negatively.

- **Recommendation**

Lock the pragma version and also consider known bugs (https://github.com/ethereum/solidity/releases) for the chosen compiler version.

Pragma statements can be allowed to float when a contract is intended for consumption by other developers, as in the case with contracts in a library or EthPM package. Otherwise, the developer would need to manually update the pragma in order to compile locally.

- **Location:**
  - Maketplace:: ALL CONTRACT
  - Staking:: ALL CONTRACT
  - Token:: ALL CONTRACT

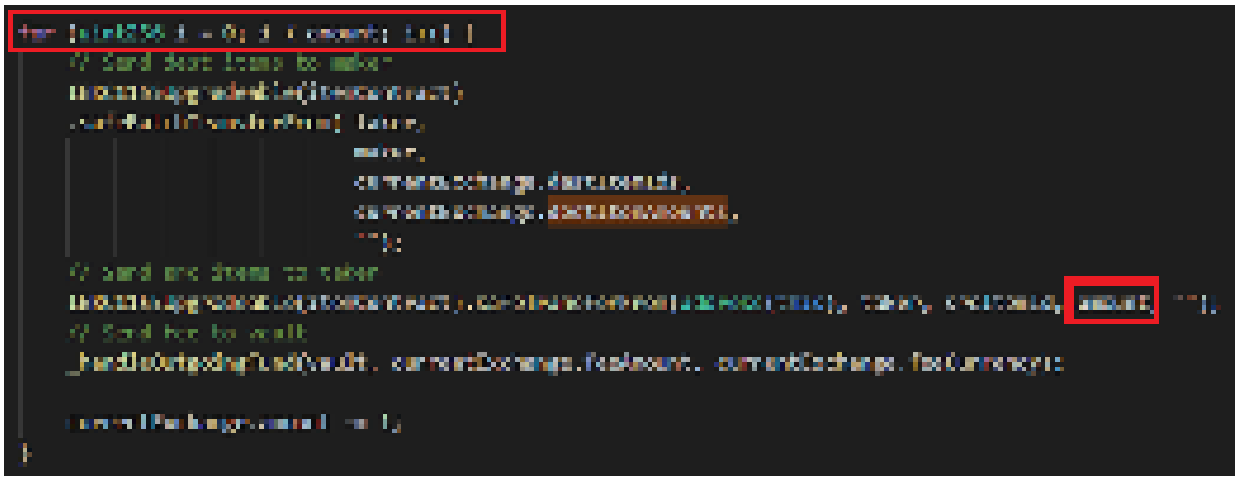
[2] Gas Optimization

1 MEDIUM

- **Overview**

Gas optimization is a matter of doing what is cheap and avoiding what is expensive in terms of gas costs on EVM blockchains.

- **Possible Impact**



```
for (uint256 i = 0; i < amount; i++) {
    [illegible]
}
```

(Blurring the image of the code snippet in the public report because the Customer's code is in the private repository)

Using a ‘*for*’ loop to transfer multiple times will cost more gas fee than transferring once. Also the handling of this '*for*' loop is wrong which leads to a logic business vulnerability (SC3).

- **Recommendation**

Consider removing the ‘*for*’ loop and replacing it with a transfer function call with the appropriate amount of items.

- **Location:**

  - Marketplace::ERC1155Marketplace.sol: (L456-L465)

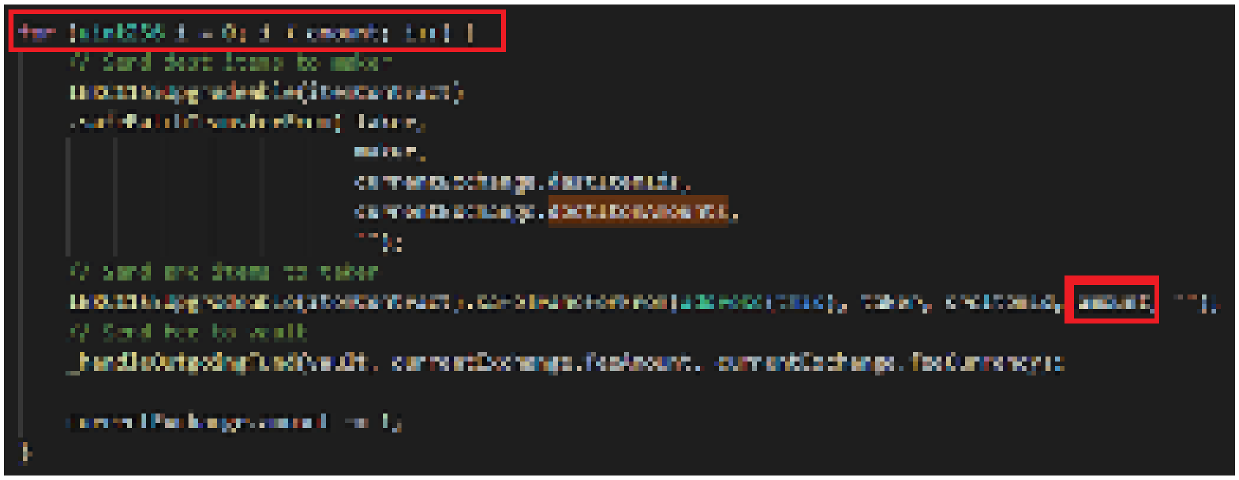
[3] Logic vulnerability

1 HIGH

- **Overview**

The smart contract will incur a loss of amount*(amount-1) items for each time the takeExchangeOffer() function is executed.

- **Possible Impact**



```
[illegible]
```

(Blurring the image of the code snippet in the public report because the Customer's code is in the private repository)

Since the maker only deposited '*amount*' items in the exchangeOffer() function, but with the execution of the 'for' loop in the takeExchangeOffer() function, the exchange transfers *amount * amount* items to taker. The attacker can act as both a maker and a taker to extract profits from the exchange.

- **Recommendation**

Consider removing the '*for*' loop, replacing it with a transfer function call with the appropriate amount of items.

- **Location:**

  - Marketplace::ERC1155Marketplace.sol: (L456-L465)

3. CONCLUSION

This document, and its appendices, represents the results of several days of our intensive work.

Smart contracts within the scope were analyzed with static analysis tools and manually reviewed.

Please feel free to direct any questions on this assessment to: audit@securichain.io.

APPENDIX 1. ASSESSMENT LIST

CHECKLIST		
Arithmetic operations		
	Integer Overflow/Underflow	Integer Division
	Integer Truncation	Integer Sign
	Wrong Operator	
Re-entrancy		
Bad Randomness		
	Timestamp Dependence	Blockhash
Front running		
DDos		
	DOS By Complex Fallback Function	DOS By Gaslimit
	DOS By Non-existent Address Or Malicious Contract	
Unsafe external calls		
Gas usage		
	Invariants in Loop	Invariants State Variables Are Not Declared Constant
Business Logics Review		
Access Control & Authorization		
	Replay Attack	Use tx.origin For Authentication
Logic Vulnerability		

APPENDIX 2. RISK RATING

Risk Level	Explain	Example Types
High	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.	Re-entrancy
		Front running
		DDos
		Bad Randomness
		Logic Vulnerability
		Arithmetic operations
Medium	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.	Access Control
		Unsafe external calls
		Business Logics Review
		Logic Vulnerability
Low	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.	Gas usage
Info	The issue does not pose an immediate risk, but is relevant to security best practices or Defence in Depth.	Do not specify a specific version of Solidity