Assure DeFi™

The Verification **Gold Standard ™**

Security Assessment

tstMMA Token

November 11, 2023

Audit Status: Pass

Audit Edition: Standard

ASSURE DEFI™

THE VERIFICATION GOLD STANDARD

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
Critical	Danger or Potential Problems.
High	Be Careful or Fail test.
Low	Pass, Not-Detected or Safe Item.
Informational	Function Detected

Manual Code Review Risk Results

Contract Privilege	Description
Buy Tax	0%
Sale Tax	0%
Cannot Sale	Pass
Cannot Sale	Pass
Max Tax	0%
Modify Tax	No
Fee Check	Pass
Is Honeypot?	Not Detected
Trading Cooldown	Not Detected
Can Pause Trade?	Fail, owner need to enable trade.
Pause Transfer?	Detected, owner need to enable trade.
Max Tx?	Pass
Is Anti Whale?	Not Detected
Is Anti Bot?	Not Detected

Contract Privilege	Description
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Not Detected
Can Mint?	Pass
Is Proxy?	Not Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	0xa101ed9019948ec99Dc5272fc9e90Dcb752Eb41e
Self Destruct?	Not Detected
External Call?	Not Detected
Other?	Not Detected
Holders	1
Auditor Confidence	Low

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview

Token Summary

Parameter	Result
Address	
Name	tstMMA
Token Tracker	tstMMA (tMMA)
Decimals	18
Supply	100,000,000
Platform	Binance Smart Chain
compiler	v0.8.20+commit.a1b79de6
Contract Name	MMATokena
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	Corporate

Main Contract Assessed Contract Name

Name	Contract	Live
tstMMA		Yes

TestNet Contract Assessed Contract Name

Name	Contract	Live
tstMMA	0xE8BfE8Ca88Db4AF097AE1a0Fe283328e69f280C0	Yes

Solidity Code Provided

SolID	File Sha-1	FileName
MMA	cd3e4bf9699d6fbc592a28b1e4a18664	tMMA.sol
MMA		
MMA		
MMA		

Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	tMMA.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	tMMA.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	tMMA.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	tMMA.sol	L: 138 C: 84
SWC-104	Pass	Unchecked Call Return Value.	tMMA.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	tMMA.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	tMMA.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	tMMA.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	tMMA.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	tMMA.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	tMMA.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	tMMA.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	tMMA.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	tMMA.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-114	Pass	Transaction Order Dependence.	tMMA.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	tMMA.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	tMMA.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	tMMA.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	tMMA.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	tMMA.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	tMMA.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	tMMA.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	tMMA.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	tMMA.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	tMMA.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	tMMA.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	tMMA.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	tMMA.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	tMMA.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	tMMA.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	tMMA.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	tMMA.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-132	Pass	Unexpected Ether balance.	tMMA.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	tMMA.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	tMMA.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	tMMA.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	tMMA.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Smart Contract Vulnerability Details

SWC-103 - Floating Pragma.

CWE-664: Improper Control of a Resource Through its Lifetime.

References:

Description:

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Remediation:

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

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.

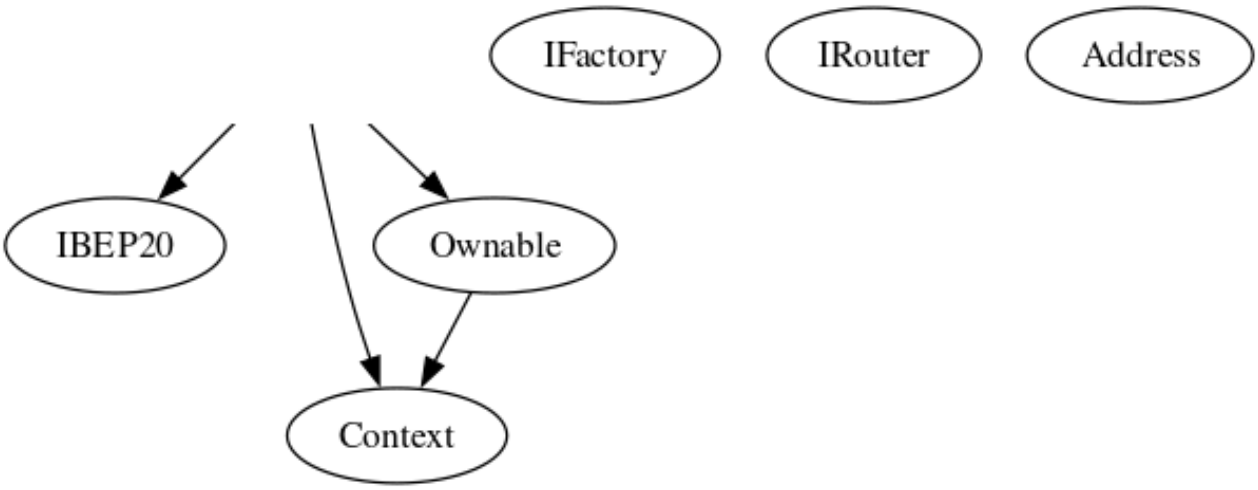
References:

Ethereum Smart Contract Best Practices - Lock pragmas to specific compiler version.

Inheritance

The contract for tstMMA has the following inheritance structure.

The Project has a Total Supply of 100,000,000



Smart Contract Advance Checks

ID	Severity	Name	Result	Status
tMMA-01	Low	Potential Sandwich Attacks.	Pass	Not Detected
tMMA-02	Informational	Function Visibility Optimization	Pass	Not Detected
tMMA-03	Low	Lack of Input Validation.	Fail	Detected
tMMA-04	High	Centralized Risk In addLiquidity.	Pass	Not Detected
tMMA-05	Low	Missing Event Emission.	Pass	Not Detected
tMMA-06	Low	Conformance with Solidity Naming Conventions.	Pass	Not Detected
tMMA-07	Low	State Variables could be Declared Constant.	Pass	Not Detected
tMMA-08	Low	Dead Code Elimination.	Pass	Not Detected
tMMA-09	High	Third Party Dependencies.	Pass	Not Detected
tMMA-10	High	Initial Token Distribution.	Pass	Not Detected
tMMA-11	High	onlyDev configured as hidden owner.	Pass	Not Detected
tMMA-12	High	Centralization Risks In The X Role	Pass	Not Detected
tMMA-13	Informational	Extra Gas Cost For User..	Pass	Not Detected
tMMA-14	Medium	Unnecessary Use Of SafeMath	Pass	Resolved
tMMA-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected
tMMA-16	Medium	Taxes can be up to 100%	Pass	Not Detected
tMMA-17	Logical Issue	Conformance to numeric notation best practice.	Pass	Not Detected
tMMA-18	Critical	Stop Transactions by using Enable Trade.	Fail	Detected

tMMA-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	Low	tMMA.sol: L: 753 C: 14, L: 758 2C: 14, L: 763 2C: 14	Detected

Description

The given input is missing the check for the non-zero address.

The given input is missing the check for the all onlyOwner..

Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...
 require(receiver != address(0), "Receiver is the zero address");
...
...
require(value X limitation, "Your not able to do this function");
...
```

We also recommend customer to review the following function that is missing a required validation. all onlyOwner..

tMMA-18 | Stop Transactions by using Enable Trade.

Category	Severity	Location	Status
Logical Issue	Critical	tMMA.sol: L: 747 C: 14	Detected

Description

Enable Trade is presend on the following contract and when combined with Exclude from fees it can be considered a whitelist process, this will allow anyone to trade before others and can represent and issue for the holders.

Remediation

We recommend the project owner to carefully review this function and avoid problems when performing both actions.

Project Action

Technical Findings Summary

Classification of Risk

Severity	Description
Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
Low	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
Critical	1	0	0
High	0	0	0
Medium	0	0	0
Low	3	2	0
Informational	1	0	0
Total	5	0	0

Social Media Checks

Social Media	URL	Result
Twitter	https://www.twitter.com/memealliancefps	Pass
Other		Fail
Website	https://www.meme-alliance.com	Pass
Telegram	https://t.me/RealElmoERC	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Assessment Results

Score Results

Review	Score
Overall Score	88/100
Auditor Score	85/100
Review by Section	**Score**
Manual Scan Score	22/53
SWC Scan Score	35 /37
Advance Check Score	31 /40

The Following Score System Has been Added to this page to help understand the value of the audit, the maximun score is 100, however to attain that value the project most pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below.

Audit Passed

Assessment Results

Important Notes:

- The contract needs optimization and fixes.ı
- The contract uses the ERC20 error libraries.

Auditor Score =85
Audit Passed

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.