ASSURE DEFI®
THE VERIFICATION GOLD STANDARD

Security Assessment:

More Pad STAKING

MOREPAD

December 18, 2024

- Audit Status: **Fail**
- Audit Edition: **Advance**

VERIFIED BY ASSURE DEFI®
INTEGRITY • TRUST • CREDIBILITY

Project Overview

Token Summary

Parameter	Result
Address	
Name	More Pad
Token Tracker	More Pad (MPAD)
Decimals	18
Supply	
Platform	BNBCHAIN
compiler	0.8.22+commit.4fc1097e
Contract Name	MOREPAD
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	Corporate

Main Contract Assessed
Contract Name

Name	Contract	Live
More Pad		Yes

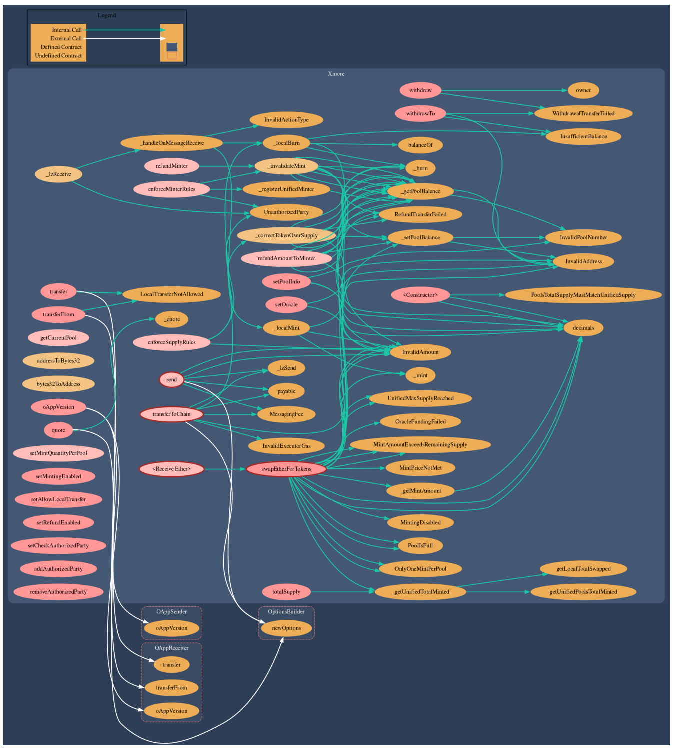
TestNet Contract was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
MOT_1	13fa00fd790e769e7c523df5df60ac04cfc0148d	MOT_1.sol
MOT_1		.sol
MOT_1		.sol
MOT_1		.sol
MOT_1		.sol
MOT_1		.sol

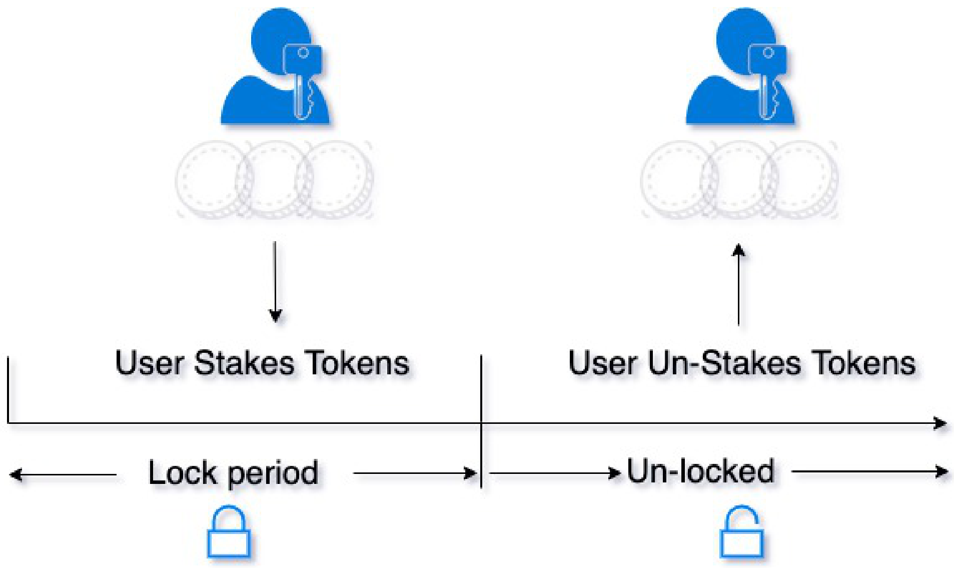
Call Graph

The contract for More Pad has the following call graph structure.



What is a Staking Contract

A smart contract which allows users to stake and un-stake a specified ERC20 token. Staked tokens are locked for a specific length of time (set by the contrat owner at the outset). Once the time period has elapsed, the user can remove their tokens again.



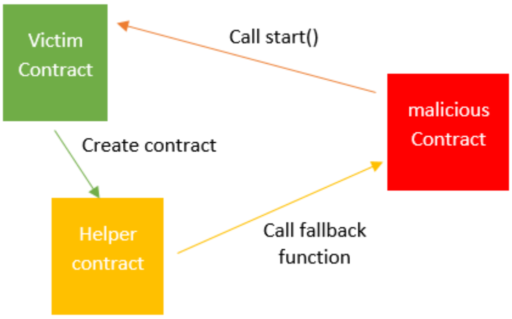
Reentrancy Check

The Project Owners of More Pad have not configure the Reentrancy Guard library.

You can read more about Reentrancy issues in the following link.
Reentrancy After Istanbul.

We recommend the team to add the library to the contract to avoid potential issues.

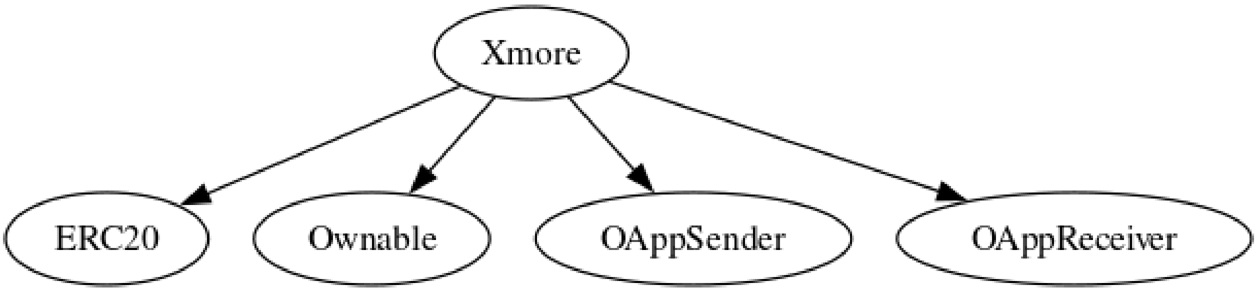
We recommend the team to create a new contract with Reentrancy Guard added to the same.



Inheritance

The contract for More Pad has the following inheritance structure.

The Project has a Total Supply of



Privileged Functions (onlyOwner)

Please Note if the contract is Renounced none of this functions can be executed.

Function Name	Parameters	Visibility
refuncdMinter		External
refundAmountToMinter		External
setMintQuantityPerPool		External
setOracle		Public
setPoolInfo		Public
setMintingEnabled		Public
setAllowLocalTransfer		Public
setRefundEnabled		Public
withdraw		Public
withdrawTo		Public
setCheckAuthorizedParty		Public
addAuthorizedParty		Public
removeAuthorizedParty		Public

MPAD-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	🟢 Low	MOT_1.sol: L: 846 C: 12, L: 850 C: 12, L: 854 C: 12, L: 858 C: 12	Detected

Description

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

The given input is missing the check for the onlyOwners need to be revisited for require..

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. onlyOwners need to be revisited for require..

MPAD-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	🟢 Low	MOT_1.sol: L: 846 C: 12, L: 850 C: 12, L: 854 C: 12, L: 858 C: 12	Detected

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.The linked code does not create an event for the transfer.

Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

MPAD-19 | Centralization Privileges of onlyOwner.

Category	Severity	Location	Status
	🟡 Medium	MOT_1.sol:	Detected

Description

Centralized Privileges are found on the functions outlined in the OnlyOwner Section.

Remediation

Inheriting from Ownable and calling its constructor on yours ensures that the address deploying your contract is registered as the owner. The onlyOwner modifier makes a function revert if not called by the address registered as the owner.

Project Action

MPAD-20 | Reentrancy Vulnerability .

Category	Severity	Location	Status
Centralization / Privilege	● Low	MOT_1.sol: L:677, L:177	Detected

Description

Potential reentrancy issues in Ether transfer functions.

Remediation

Use reentrancy guards and ensure state changes precede external calls.

Project Action

MPAD-21 | Incorrect Pool Number Handling.

Category	Severity	Location	Status
Input Validation	🟡 Medium	MOT_1.sol:	Detected

Description

Pool numbers are not properly validated, potentially causing incorrect operations.

Remediation

Add checks to ensure pool numbers are within valid range.

Project Action

MPAD-22 | Unauthorized Access to Oracle Functions.

Category	Severity	Location	Status
Access Control	🟠 High	MOT_1.sol:	Detected

Description

Oracle-related functions can be called by unauthorized addresses.

Remediation

Ensure only the designated oracle address can call these functions.

Project Action

MPAD-23 | Oracle Fee Mismanagement.

Category	Severity	Location	Status
Financial	● Critical	MOT_1.sol:	Detected

Description

Oracle fee handling may lead to incorrect fund transfers.

Remediation

Validate oracle fee logic and ensure correct fund allocation.

Project Action

MPAD-24 | Refund Transfer Failure.

Category	Severity	Location	Status
Error Handling	● Informational	MOT_1.sol:	Detected

Description

Refunds may fail due to insufficient gas or balance.

Remediation

Implement a more robust refund mechanism with event logging for failures.

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.
ℹ 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	1	0
🟠 High	1	1	0
🟡 Medium	3	3	0
🟢 Low	2	2	0
ℹ Informational	1	0	0
Total	8	8	0

Social Media Checks

Social Media	URL	Result
Twitter	https://x.com/morepad404	Pass
Other		N/A
Website	https://www.morepad.io/	Pass
Telegram		Fail

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	75/100
Auditor Score	80/100

Review by Section	Score
Manual Scan Score	25
Auto Scan Score	37
Advance Check Score	13

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum 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 Fail

Assessment Results

Important Notes:

- Contract Initialization: Ensure constructor parameters are validated. Check if unifiedMaxSupply and poolsInfo are correctly set.ı
- Access Control: Verify onlyOwner modifiers are used appropriately. Ensure authorizedParties are correctly managed.ı
- Minting Logic: Validate minting conditions in swapEtherForTokens. Ensure _getMintAmount calculations are accurate.ı
- Ether Handling: Check for correct Ether transfer logic in refunds and withdrawals. Ensure oracleFee is properly deducted and transferred.ı
- Cross-Chain Operations: Validate LayerZero message sending and receiving. Ensure _handleOnMessageReceive processes messages securely.ı
- State Management: Confirm pool balances and limits are enforced. Validate logic in _correctTokenOverSupply.ı
- Reentrancy: Ensure no reentrancy vulnerabilities, especially in Ether transfers.ı
- Oracle Security: Verify oracle updates and access control. Ensure enforceMinterRules and enforceSupplyRules are secure.ı
- Fallback Function: Ensure receive() function is secure and correctly processes Ether.ı

- Testing: Conduct thorough unit and integration tests. Test edge cases for minting, refunds, and cross-chain transfers.

Auditor Score =80
Audit Fail

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.

ASSURE DEFI™
THE VERIFICATION **GOLD STANDARD**