AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

TABLE OF CONTENTS

contact@auditmos.com

AUDITMOS

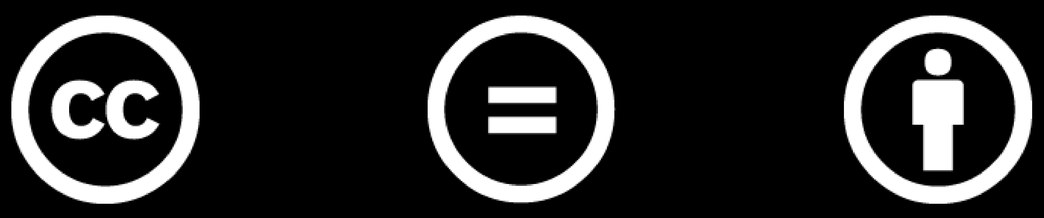
Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

LICENSE



contact@auditmos.com

AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

DISCLAIMER

contact@auditmos.com

AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

APPROACH AND METHODOLOGY

PURPOSE

1. Determine the correct operation of the protocol, according to the design specification.
2. Identify possible vulnerabilities that could be exploited by an attacker.
3. Detect errors in the smart contract that could lead to unexpected behavior.
4. Analyze whether best practices were followed during development.
5. Make recommendations to improve security and code readability.

CODEBASE

Repository	https://basescan.org/address/0xbdf317f9c153246c429f23f4093087164b145390#code
Branch	-
Commit hash	-

METHODOLOGY

1. Reading the available documentation and understanding the code.
2. Doing automated code analysis and reviewing dependencies.
3. Checking manually source code line by line for security vulnerabilities.
4. Following guildlines and recommendations.
5. Preparing this report.

contact@auditmos.com

AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

DESCRIPTION

Issues Categories:

Severity	Description
CRITICAL	vulnerability that can lead to loss of funds, failure to recover blocked funds, or catastrophic denial of service.
HIGH	vulnerability that can lead to incorrect contract state or unpredictable operation of the contract.
MEDIUM	failure to adhere to best practices, incorrect usage of primitives, without major impact on security.
LOW	recommendations or potential optimizations which can lead to better user experience or readability.

Each issue can be in the following state:

State	Description
PENDING	still waiting for resolving
ACKNOWLEDGED	know but not planned to resolve for some reasons
RESOLVED	fixed and deployed

contact@auditmos.com

AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

AUDIT SCOPE

1. getting to know the project	✓
2. research into architecture	✓
3. manual code read	✓
4. permissions of state changing functions	✓
5. identify common Solidity vulnerabilities	✓
6. test coverage	✓
7. static analysis	✓
8. storage key overlaps	✓
9. DOS possibilities by malicious attacker	✓
10. steal funds possibilities	✓

contact@auditmos.com

AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

FINDINGS

Finding	Severity	Status
#1 - Initial Token Distribution Centralization Risk	LOW	ACKNOWLEDGED
#2 - Front-Running Vulnerability in EIP-2612 Permit	LOW	ACKNOWLEDGED
#3 - Missing Token Recovery Mechanism	LOW	ACKNOWLEDGED

contact@auditmos.com

AUDITMOS

Secure your space

AI AGENT LAYER

audit / code review report

November 08, 2024

#1 - INITIAL TOKEN DISTRIBUTION CENTRALIZATION

Severity	Status
LOW	ACKNOWLEDGED

RISK

All tokens are minted to a single address (contract deployer) in the constructor. If this address is compromised, it could lead to market manipulation or token dumping.

```
constructor() {
    _mint(msg.sender, 500_000_000_000000000000000000);
}
```

RECOMMENDATION

- Implement a vesting schedule for initial token distribution
- Use a time-lock contract for team/founder allocations
- Consider splitting initial allocation between multiple secure addresses
- Add maximum transfer limits per transaction/time period

contact@auditmos.com

AI AGENT LAYER

audit / code review report

November 08, 2024

#2 - FRONT-RUNNING VULNERABILITY IN EIP-2612 PERMIT

Severity	Status
LOW	ACKNOWLEDGED

The inherited ERC20 implementation includes EIP-2612 permit functionality. This implementation could be vulnerable to front-running attacks where an attacker could extract MEV (Maximal Extractable Value) by manipulating transaction ordering.

RECOMMENDATION

- Implement deadline checks for permit operations
- Consider adding min/max validity periods for permits
- Add nonce tracking per token approval
- Consider implementing permit2 from Uniswap for improved security

AI AGENT LAYER

audit / code review report

November 08, 2024

#3 - MISSING TOKEN RECOVERY MECHANISM

Severity	Status
LOW	ACKNOWLEDGED

If other ERC20 tokens are accidentally sent to this token contract address, they will be permanently locked as there's no mechanism to recover them. This is a common issue that has led to the loss of funds in many DeFi projects.

Impact:

- Permanent loss of any ERC20 tokens accidentally sent to the contract
- No way to recover funds in emergency situations
- Could affect both users and protocol treasury management

RECOMMENDATION

One of the possible approach is to use safeTransfer from OpenZeppelin's SafeERC20 library. Here's why:

1. Some tokens (like USDT) don't follow the ERC20 standard strictly and may return false instead of reverting on failure
2. Some tokens might not return any value at all
3. Low-level transfer calls might fail silently

auditmos.com

Secure your space

contact@auditmos.com