AUDITMOS

Secure your space

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

TABLE OF CONTENTS

contact@auditmos.com

AUDITMOS

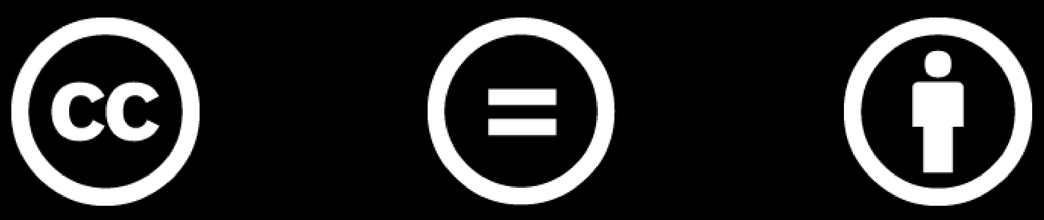
Secure your space

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

LICENSE



contact@auditmos.com

AUDITMOS

Secure your space

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

DISCLAIMER

contact@auditmos.com

AUDITMOS
Secure your space

ALASKA GOLD RUSH

audit / code review report

April 15, 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	Auditmos received zipped repository to audit
Branch	NA
Commit hash	NA

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

ALASKA GOLD RUSH

audit / code review report

April 15, 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

ALASKA GOLD RUSH

audit / code review report

April 15, 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

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

FINDINGS

Finding	Severity	Status
#1 - Wrong implementation of EIP-712	MEDIUM	RESOLVED
#2 - Some users can send messages for free	LOW	RESOLVED
#3 - Lack of validation of receiver address	LOW	RESOLVED
#4 - Redundant error	LOW	RESOLVED

contact@auditmos.com

AUDITMOS

Secure your space

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

Severity	Status
MEDIUM	RESOLVED

#1 - WRONG IMPLEMENTATION OF EIP-712

The hash structures BLOCK_FUNCTION_HASH and UNBLOCK_FUNCTION_HASH are incorrect implemented because they contain an additional argument bytes signature, which is not used during hashing. The only arguments that the structs should have are bytes32 operationId, uint256 chainId, address user, uint256 amount and uint256 blockDeadline. Everything else is not included during hashing.

Additionally, the hash structures and domain separator should use double apostrophes " instead of single ones ' by definition of EIP-712.

RECOMMENDATION

Make the following changes:

Instead of:

'EIP712Domain(string name,string version,uint256 chainId,address verifyingContract)'

'blockCarat(bytes32 operationId_,uint256 chainId_,address user_,uint256 amount_,uint256 blockDeadline_,bytes signature_)'

'unblockCarat(bytes32 operationId_,uint256 chainId_,address user_,uint256 amount_,uint256 blockDeadline_,bytes signature_)'

Use:

"EIP712Domain(string name,string version,uint256 chainId,address verifyingContract)"

"blockCarat(bytes32 operationId_,uint256 chainId_,address user_,uint256 amount_,uint256 blockDeadline_)"

"unblockCarat(bytes32 operationId_,uint256 chainId_,address user_,uint256 amount_,uint256 blockDeadline_)"

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

#2 - SOME USERS CAN SEND MESSAGES FOR FREE

Severity	Status
LOW	RESOLVED

Before sending a message to the Router contract, a fee is estimated to pay for the transaction of the destination contract. Users have to pay at least the fee if they do not want their transaction to revert, so msg.value >= fee. The excess between msg.value and fee will remain in the contract.

At some point in time, after sending a lot of messages, the contract balance will increase due to the compounding of the remaining amount, and the amount in the contract balance will be enough to send message. Some users may benefit from this and can send message for free, as the fee will be paid from the contract.

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

#3 - LACK OF VALIDATION OF RECEIVER ADDRESS

Severity	Status
LOW	RESOLVED

The MultichainTransferBase lacks validation of the receiver address. Users may mistakenly set the receiver address to be equal to address(this) or address(0).

Most popular bridges have implemented checks for the receiver address to avoid sending tokens to the wrong address.

contact@auditmos.com

AUDITMOS

Secure your space

ALASKA GOLD RUSH

audit / code review report

April 15, 2024

#4 - REDUNDANT ERROR

The CaratBlocker contract contains redundant error ZeroAddress().

Severity	Status
LOW	RESOLVED

contact@auditmos.com

auditmos.com

Secure your space

contact@auditmos.com