AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

TABLE OF CONTENTS

contact@auditmos.com

AUDITMOS

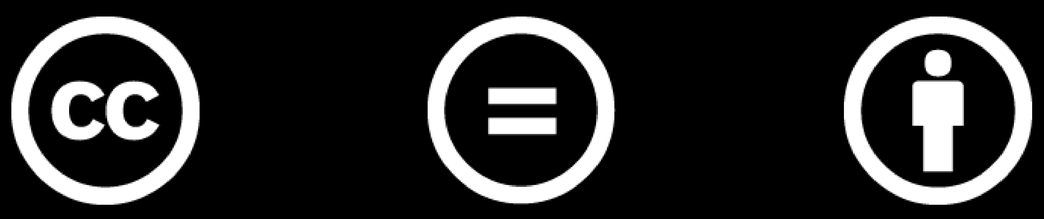
Secure your space

MINTDAO

audit / code review report

December 05, 2021

LICENSE



contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

DISCLAIMER

contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

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://github.com/mintdao/initial-liquidity-orderbook-contract/
Branch	main
Commit hash	b17e4be36edcc5ebc2885d6b32360df251597438

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

MINTDAO

audit / code review report

December 05, 2021

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

MINTDAO

audit / code review report

December 05, 2021

FINDINGS

Finding	Severity	Status
#1 - change contract name	LOW	RESOLVED
#2 - gas optimisation in STATE.load	LOW	RESOLVED
#3 - merge state_helpers.rs and state.rs	LOW	RESOLVED
#4 - better logging	LOW	RESOLVED
#5 - no need to use different schema	LOW	RESOLVED
#6 - double check permission	LOW	RESOLVED
#7 - verification of logic	LOW	RESOLVED
#8 - general feedback and good practices	LOW	RESOLVED

contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

#1 - CHANGE CONTRACT NAME

Severity	Status
LOW	RESOLVED

It is recommended to change the value of CONTRACT_NAME to better distinguish contract name from others especially when there is a need to publish contract crate on crates.io.

RECOMMENDATION

Please consider to change the current code:

```
const CONTRACT_NAME: &str = "crates.io:mint";
```

to this one below:

```
const CONTRACT_NAME: &str = "crates.io:mintdao-mint";
```

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/b17e4be36edcc5ebc2885d6b32360df251597438/src/contract.rs#L20

contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

#2 - GAS OPTIMISATION IN STATE.LOAD

Some of functions calls STATE.load twice as assert_state calls it always and some functions requires it additionally.

Severity	Status
LOW	RESOLVED

RECOMMENDATION

Please consider to change the current code:

```
assert_state(deps: &DepsMut, state: ContractState)
```

to this one below:

```
assert_state(state: &State, expected_state: ContractState)
```

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/44c39ef885f934d4e99a60b639d64e569c09fff8/src/utils.rs#L28

contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

#3 - MERGE STATE_HELPERS.RS AND STATE.RS

It is worth to consider merging state_helpers.rs and state.rs as it will be only about 120 lines of code in total.

In implemented smart contract logic some state storage shouldn't be changed outside helper functions, so it would be better to make them private for state.rs file for example state variables used in update_order.

Severity	Status
LOW	RESOLVED

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/44c39ef885f934d4e99a60b639d64e569c09fff8/src/state_helpers.rs#L39

contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

#4 - BETTER LOGGING

It is highly recommended to add more logging to some below functions:

- try_update_order (at least order identificator)
- try_claim_tokens_and_ust (amount which user is sending)

Severity	Status
LOW	RESOLVED

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/6e77c91a87b1492f72a34785f46662f7c51b2187/src/execute.rs#L52

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/6e77c91a87b1492f72a34785f46662f7c51b2187/src/execute.rs#L297

contact@auditmos.com

MINTDAO

audit / code review report

December 05, 2021

#5 - NO NEED TO USE DIFFERENT SCHEMA

There is no need to use different schema for try_update_order and try_delete_order for loading order from state.

Severity	Status
LOW	RESOLVED

RECOMMENDATION

Please consider to change the current code:

```
match get_order(deps.storage, id) {
Ok(ord) => {
order = ord;
}
_ => {
return Err(ContractError::OrderDoesNotExist);
}
}
```

to this one below:

```
let mut order = get_order(deps.storage, id).map_err(|_|
ContractError::OrderDoesNotExist)?;
```

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/6e77c91a87b1492f72a34785f46662f7c51b2187/src/execute.rs#L52

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/6e77c91a87b1492f72a34785f46662f7c51b2187/src/execute.rs#L70

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

#6 - DOUBLE CHECK PERMISSION

In try_update_config checking permission is done twice.

Severity	Status
LOW	RESOLVED

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/b17e4be36edcc5ebc2885d6b32360df251597438/src/contract.rs#L73

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/6e77c91a87b1492f72a34785f46662f7c51b2187/src/execute.rs#L118

contact@auditmos.com

AUDITMOS

Secure your space

MINTDAO

audit / code review report

December 05, 2021

#7 - VERIFICATION OF LOGIC

Please make sure if the below logic implemented in try_shuffle_orders is correct.

```
counter = counter + Uint128::from(1u128);
index = shuffled_orders_count + counter
```

Severity	Status
LOW	RESOLVED

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/6e77c91a87b1492f72a34785f46662f7c51b2187/src/execute.rs#L213

contact@auditmos.com

MINTDAO

audit / code review report

December 05, 2021

#8 - GENERAL FEEDBACK AND GOOD PRACTICES

Severity	Status
LOW	RESOLVED

It is better to use u128 instead of Uint128 when you calculating indexes - this is cheaper for save in state and doesn't require that much unnecessary code (like for e.g creating instance of Uint128 each time during the incrementation).

Uint128 is useful when when there is a need to count balances of cw20, because it parses into *.json as a string (so it is not possible that balance exceeds maximum JavaScript number).

PROOF OF SOURCE

https://github.com/mintdao/initial-liquidity-orderbook-contract/blob/44c39ef885f934d4e99a60b639d64e569c09fff8/src/state_helpers.rs#L14

auditmos.com

Secure your space

contact@auditmos.com