Museum of Mahomes Security Review

Pashov Audit Group

Conducted by: pashov

September 14th, 2023

Contents

1. About pashov

Krum Pashov, or **pashov**, is an independent smart contract security researcher. Having found numerous security vulnerabilities in various protocols, he does his best to contribute to the blockchain ecosystem and its protocols by putting time and effort into security research & reviews. Check his previous work here or reach out on Twitter @pashovkrum.

2. Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where I try to find as many vulnerabilities as possible. I can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

3. Introduction

A time-boxed security review of the **Museum of Mahomes** protocol was done by **pashov**, with a focus on the security aspects of the application's smart contracts implementation.

4. About Museum of Mahomes

The Museum of Mahomes protocol is an NFT collection with a few special features, namely burning a token to redeem the physical copy of the art, as well as an on-chain "reveal" mechanism.

More docs

Observations

Max supply of tokens is 3090 instead of the usual 10000. Owner of the contract can update the `price` at any time to any value.

The `DelegationRegistry` is an attack vector as it manages token allowances for revealing and redeeming. It is ouf of scope for this audit.

Privileged Roles & Actors

- Collection owner - can claim the mint funds as well as transfer ownership, set mint price and set an address to be a `treasury` one
- Collection metadata owner - controls the `revealOpen`, `redeemOpen` and the `baseURI` properties
- Treasury account - can mint NFTs for free
- Delegation Registry - manages token delegations, which are basically allowances for revealing and redeeming (burning) NFTs
- Minter - can pay ETH to mint NFTs

5. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

5.1. Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

5.2. Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5.3. Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

6. Security Assessment Summary

review commit hash **- c63f28585f73b94a556bdce45831bbcd017790e3**

fixes review commit hash **- e4d0115b931c31e0dcd92433e54aba0f1b09ec7f**

Scope

The following smart contracts were in scope of the audit:

- `MuseumOfMahomes`

7. Executive Summary

Over the course of the security review, pashov engaged with Museum of Mahomes to review Museum of Mahomes. In this period of time a total of **6** issues were uncovered.

Protocol Summary

Protocol Name	Museum of Mahomes
Date	September 14th, 2023

Findings Count

Severity	**Amount**
High	1
Low	5
Total Findings	**6**

Summary of Findings

ID	Title	Severity	Status
[H-01]	Last NFT from the supply can't be minted	High	Resolved
[L-01]	Reveal and Redeem should only be set to true	Low	Resolved
[L-02]	All state-changing methods should emit events	Low	Resolved
[L-03]	A treasury account can mint all NFTs	Low	Resolved
[L-04]	Contract is not working as a state machine	Low	Resolved
[L-05]	Use a two-step access control transfer pattern	Low	Resolved

8. Findings

8.1. High Findings

[H-01] Last NFT from the supply can't be minted

Severity

Impact: Medium, as only one NFT won't be available for minting, but this is value loss to the protocol

Likelihood: High, as it's impossible to mint the last NFT

Description

Currently both the `mint` and `mintPhysical` methods have the following check:

```
if (nextId + amount >= MAX_SUPPLY) revert ExceedsMaxSupply();
```

This is incorrect, as even when the `nextId` is `MAX_SUPPLY - 1` then an `amount` of 1 should be allowed but with the current check the code will revert. This is due to the `equal` sign in the check, which shouldn't be there. Here is a Proof of Concept unit test demonstrating the issue (add it to `MuseumOfMahomes.t.sol`):

```
function testNotAllNFTsCanBeMinted() public {
        museum.setPrice(PRICE);
        uint256 allButOneNFTSupply = 3089;

        // mint all but one from the NFT `MAX_SUPPLY` (3090)
        museum.mint{value: allButOneNFTSupply * PRICE}(address
          (this), allButOneNFTSupply);
        require(allButOneNFTSupply == museum.balanceOf(address
          (this)), "Mint did not work");

        // try to mint the last NFT from the supply, but it doesn't work
        vm.expectRevert(MuseumOfMahomes.ExceedsMaxSupply.selector);
        museum.mint{value: PRICE}(address(this), 1);
    }
```

Recommendations

Do the following change in both `mint` and `mintPhysical`:

```
- if (nextId + amount >= MAX_SUPPLY) revert ExceedsMaxSupply();
+ if (nextId + amount > MAX_SUPPLY) revert ExceedsMaxSupply();
```

8.2. Low Findings

[L-01] Reveal and Redeem should only be set to `true`

Currently the `setRevealOpen` and `setRedeemOpen` methods allow setting the values to both `true` and `false` as many times as the `metadataOwner` decides to. This shouldn't be the case, as both should only be available to set to `true` just once, and never to `false` after this. Change the setters to methods that only set the values to `true`, removing the parameters from the methods.

[L-02] All state-changing methods should emit events

Currently most of the state-changing methods in the `MuseumOfMahomes` contract do not emit an event. An example is the `setPrice` method, which might be important for users or front-end/UI clients that wish to monitor and track the current price of the NFTs. Add proper event emissions in all state-changing methods.

[L-03] A `treasury` account can mint all NFTs

Currently an account that is in the `treasury` mapping can mint all NFTs for free. While it is desired that such an account does not pay for minting a token, consider adding a `MAX_TREASURY_MINTS` upper bound to limit the count of NFTs minted by `treasury` accounts. You can also make sure that when a `treasury` account is minting, the `msg.value` is 0.

[L-04] Contract is not working as a state machine

Currently it is possible for the `metadataOwner` to set the `redeemOpen` value to `true` while the `revealOpen` hasn't been set to `true` yet. There should be a sequence/flow of how the contract works - first minting, then revealing, then redeem (or redeem right after reveal). Allow setting `redeemOpen` to `true` only if `revealOpen == true`, and also allow setting `revealOpen` to `true` only when mint is completed (`totalSupply == MAX_SUPPLY`).

[L-05] Use a two-step access control transfer pattern

The `MuseumOfMahomes` contract uses a single-step access control transfer pattern in `setOwner` and `setMetadataOwner`. This means that if the current `owner` or `metadataOwner` accounts call the methods with an incorrect address, then those roles will be lost forever along with all the functionality that depends on them. Follow the pattern from OpenZeppelin's Ownable2Step and implement a two-step transfer pattern for the actions.