Baba Marta Audit Report

Prepared by: Thembinkosi Mkhonta

Table of Contents

Protocol Summary

The "Baba Marta" protocol allows you to buy `MartenitsaToken` and to give it away to friends. Also, if you want, you can be a producer. The producer creates `MartenitsaTokens` and sells them. There is also a voting for the best `MartenitsaToken`. Only producers can participate with their own `MartenitsaTokens`. The other users can only vote. The winner wins 1 `HealthToken`. If you are not a producer and you want a `HealthToken`, you can receive one if you have 3 different `MartenitsaTokens`. More `MartenitsaTokens` more `HealthTokens`. The `HealthToken` is a ticket to a special event (producers are not able to participate). During this event each participant has producer role and can create and sell own `MartenitsaTokens`.

Disclaimer

This audit is not yet complete

Risk Classification

		Impact		
		High	Medium	Low
	High	H	H/M	M
Likelihood	Medium	H/M	M	M/L
	Low	M	M/L	L

We use the CodeHawks severity matrix to determine severity. See the documentation for more details.

Audit Details

The findings described in this document correspond the following commit hash:

```

```

Scope

```
./src/
├── src
│   ├── HealthToken.sol
│   ├── MartenitsaEvent.sol
│   ├── MartenitsaMarketplace.sol
│   ├── MartenitsaToken.sol
│   ├── MartenitsaVoting.sol
│   ├── SpecialMartenitsaToken.sol
```

Roles

Producer - Should be able to create martenitsa and sell it. The producer can also buy martenitsa, make present and participate in vote. The martenitsa of producer can be candidate for the winner of voting.

User - Should be able to buy martenitsa and make a present to someone else. The user can collect martenitsa tokens and for every 3 different martenitsa tokens will receive 1 health token. The user is also able to participate in a special event and to vote for one of the producer's martenitsa.

Executive Summary

Issues found

Severity	Number of issues found
High	1
Medium	2
Low	4
Info	0
Total	7

Findings

High

[H-1] Arbitrary `from` passed to `transferFrom` (or `safeTransferFrom`)

Description: Passing an arbitrary `from` address to `transfer` (or `safeTransferFrom` can lead to loss of funds, because anyone can tranfer tokens from the `from` address if an approval is made)

- Found in src/MartenitsaMarketplace.sol Line: 82

```
    martenitsaToken.safeTransferFrom(seller, buyer, tokenId);
```

Impact:

Proof of Concept:

Recommended Mitigation:

Medium

[M-1] Centralization Risk for trusted owners

Description: Contracts have owners with privileged rights to perform admin tasks and need to be trusted to not perform malicious updates.

- Found in src/HealthToken.sol Line: 10

```
contract HealthToken is ERC20, Ownable {
```

- Found in src/HealthToken.sol Line: 19

```
    function setMarketAndVotingAddress(address martenitsaMarketplace,
address martenitsaVoting) public onlyOwner {
```

- Found in src/MartenitsaEvent.sol Line: 22

```
    constructor(address healthToken) onlyOwner {
```

- Found in src/MartenitsaEvent.sol Line: 30

```
    function startEvent(uint256 duration) external onlyOwner {
```

- Found in src/MartenitsaEvent.sol Line: 60

```
    function stopEvent() external onlyOwner {
```

- Found in src/MartenitsaMarketplace.sol Line: 8

```
contract MartenitsaMarketplace is Ownable {
```

- Found in src/MartenitsaToken.sol Line: 7

```
contract MartenitsaToken is ERC721, Ownable {
```

- Found in src/MartenitsaToken.sol Line: 24

```
    function setProducers(address[] memory _producersList) public
onlyOwner{
```

- Found in src/MartenitsaVoting.sol Line: 8

```
contract MartenitsaVoting is Ownable {
```

- Found in src/MartenitsaVoting.sol Line: 34

```
    function startVoting() public onlyOwner {
```

- Found in src/MartenitsaVoting.sol Line: 57

```
    function announceWinner() external onlyOwner {
```

Impact: Contracts have owners with privileged rights to perform admin tasks and need to be trusted to not perform malicious updates or drain funds.

[M-2] Using `ERC721::_mint()` can be dangerous

Description: Using `ERC721::_mint()` can mint ERC721 tokens to addresses which do not support ERC721 tokens. Use `_safeMint()` instead of `_mint()` for ERC721.

- Found in src/HealthToken.sol Line: 32

```

```

```
    _mint(to, amountToMint);
```

Recommended Mitigation:

Change the `_mint()` function and use `_safeMint()` instead.

```
-    _mint(to, amountToMint);
+    _safeMint(to, amountToMint);
```

Low

[L-1] Unsafe ERC20 Operations should not be used

Description: ERC20 functions may not behave as expected. For example: return values are not always meaningful.

- Found in src/MartenitsaEvent.sol Line: 52

```
    (bool success) = _healthToken.transferFrom(msg.sender,
address(this), healthTokenRequirement);
```

Recommended Mitigation: It is recommended to use OpenZeppelin's SafeERC20 library.

[L-2] Solidity pragma should be specific, not wide

Description:

- Found in src/HealthToken.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaEvent.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaMarketplace.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaToken.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaVoting.sol Line: 2

```
pragma solidity ^0.8.21;
```

Recommended Mititgation: Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

[L-3] PUSH0 is not supported by all chains

Description: Solc compiler version 0.8.20 switches the default target EVM version to Shanghai, which means that the generated bytecode will include PUSH0 opcodes.

- Found in src/HealthToken.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaEvent.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaMarketplace.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaToken.sol Line: 2

```
pragma solidity ^0.8.21;
```

- Found in src/MartenitsaVoting.sol Line: 2

```
pragma solidity ^0.8.21;
```

Recommended Mitigation: Be sure to select the appropriate EVM version in case you intend to deploy on a chain other than mainnet like L2 chains that may not support PUSH0, otherwise deployment of your contracts will fail.

Informational

[I-1] Missing checks for `address(0)` when assigning values to address state variables

Description:

- Found in src/HealthToken.sol Line: 20

```
        _martenitsaMarketplace =
MartenitsaMarketplace(martenitsaMarketplace);
```

- Found in src/HealthToken.sol Line: 21

```
        _martenitsaVoting = MartenitsaVoting(martenitsaVoting);
```

- Found in src/MartenitsaEvent.sol Line: 23

```
        _healthToken = HealthToken(healthToken);
```

- Found in src/MartenitsaMarketplace.sol Line: 30

```
        healthToken = HealthToken(_healthToken);
```

- Found in src/MartenitsaMarketplace.sol Line: 31

```
        martenitsaToken = MartenitsaToken(_martenitsaToken);
```

- Found in src/MartenitsaVoting.sol Line: 27

```
        _martenitsaMarketplace = MartenitsaMarketplace(marketplace);
```

- Found in src/MartenitsaVoting.sol Line: 28

```
        _healthToken = HealthToken(healthToken);
```

Recommended Mitigation: Check for `address(0)` when assigning values to address state variables.

[I-2] `public` functions not used internally could be marked `external`

- Found in src/HealthToken.sol Line: 19

```
    function setMarketAndVotingAddress(address martenitsaMarketplace,
address martenitsaVoting) public onlyOwner {
```

- Found in src/MartenitsaToken.sol Line: 24

```
    function setProducers(address[] memory _producersList) public
onlyOwner{
```

- Found in src/MartenitsaVoting.sol Line: 34

```
    function startVoting() public onlyOwner {
```

Recommended Mitigation: Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

[I-3] Internal functions called only once can be inlined

Instead of separating the logic into a separate function, consider inlining the logic into the calling function. This can reduce the number of function calls and improve readability.

- Found in src/MartenitsaEvent.sol Line: 78

```
    function _addProducer(address _producer) internal {
```

[I-4] Unchanged variables should be constant or immutable

Constant instances:

```
MartenitsaEvent.healthTokenRequirement (src/MartenitsaEvent.sol#14) should
be constant
MartenitsaMarketplace.requiredMartenitsaTokens
(src/MartenitsaMarketplace.sol#13) should be constant
MartenitsaVoting.duration (src/MartenitsaVoting.sol#15) should be constant
```

Immutable instances:

```
MartenitsaEvent._healthToken (src/MartenitsaEvent.sol#9) should be
immutable
MartenitsaMarketplace.healthToken (src/MartenitsaMarketplace.sol#10) should
```

```
be immutable
MartenitsaMarketplace.martenitsaToken (src/MartenitsaMarketplace.sol#11)
should be immutable
MartenitsaVoting._healthToken (src/MartenitsaVoting.sol#12) should be
immutable
MartenitsaVoting._martenitsaMarketplace (src/MartenitsaVoting.sol#10)
should be immutable
```