Possum Audit Report

Version 1.0

Mahdi Rostami

November 12, 2024

Possum Passel Market Audit Report

Mahdi Rostami

November 12, 2024

Prepared by: Mahdi Rostami [Twitter]

Table of Contents

Possum Passel Market Security Review

A security review of the Passel Market, GitHub smart contracts was done by mahdi rostami .
This audit report includes all the vulnerabilities, issues and code improvements found during the security review.

Disclaimer

"Audits are a time, resource and expertise bound effort where trained experts evaluate smart contracts using a combination of automated and manual techniques to find as many vulnerabilities as possible. Audits can show the presence of vulnerabilities **but not their absence**."

- Secureum

Impact

- **High** Issues that lead to the loss of user funds. Such issues include:
  - Direct theft of any user funds, whether at rest or in motion.
  - Long-term freezing of user funds. Theft or long term freezing of unclaimed yield or other assets.
  - Protocol insolvency
- **Medium** Issues that lead to an economic loss but do not lead to direct loss of on-chain assets. Examples are:
  - Gas griefing attacks (make users overpay for gas)
  - Attacks that make essential functionality of the contracts temporarily unusable or inaccessible.
  - Short-term freezing of user funds.
- **Low** Issues where the behavior of the contracts differs from the intended behavior (as described in the docs and by common sense), but no funds are at risk.

Likelihood

- **High**
  - attack path is possible with reasonable assumptions that mimic on-chain conditions and the cost of the attack is relatively low to the amount of funds that can be stolen or lost.

- **Medium**
  - only conditionally incentivized attack vector, but still relatively likely.
- **Low**
  - has too many or too unlikely assumptions or requires a huge stake by the attacker with little or no incentive.

Actions required by severity level

- **High** - client **must** fix the issue.
- **Medium** - client **should** fix the issue.
- **Low** - client **could** fix the issue.

Executive summary

Overview

Project Name	Kintsu
Repository	Link
Commit Hash	b213104
Docs	No docs
Methods	Manual Review

Scope

File
PasselMarket

Compatibilities

- Solc Version: 0.8.19
- Chain(s) to deploy contract to: Arbitrum

Known Issues

None

Issues found

Severity	Count
High	0
Medium	0
Low	0
Informational	7

Findings

Informational

Unused Custom Error

it is recommended that the definition be removed when custom error is unused

2 Found Instances:

- Found in PasselExplorer

```
error IsRevoked();
```

- Found in PasselNFT

```
error CollectionIncomplete();
```

Check nft existance with totalsupply

2 Found Instances:

- Found in PasselExplorer

```
if (PASSEL_NFT.ownerOf(_tokenID) == address(0)) {}
if (PASSEL_NFT.ownerOf(_tokenID) == address(0)) {}
```

Unused Imports

Redundant import statement. Consider removing it.

1 Found Instances

- Found in PasselQuests

```solidity
import {IERC721} from "@openzeppelin/contracts/token/ERC721/IERC721.sol";
```

Unused mapping

Redundant import statement. Consider removing it.

1 Found Instances

- Found in PasselQuests

```solidity
mapping(uint256 nftID => uint256[] quests) public getCompletedQuests;
```

quest 0 not exist and bypassed in `PasselQuests::quest`

~~~
```solidity
if (_questID > QUESTS_AVAILABLE) revert InvalidQuest();
```
~~~

Boolean equality is not required.

3 Found Instances

- Found in PasselQuests

```solidity
if (mintingDisabled == true) revert MintingDisabled(); //@audit true not needed
if (isListedForSale[_tokenID] == false) revert NotListedForSale(); //@audit false not needed, use !isListedForSale[_tokenID]
if (_getListed != isListedForSale[_tokenID]) isListedForSale[_tokenID] = _getListed; //@audit better way: if (!isListedForSale[_tokenID]) isListedForSale[_tokenID] = true;
```

Modifiers invoked only once can be shoe-horned into the function

1 Found Instances

- Found in PasselQuests

```
modifier onlyExplorer()
```