Client Firm: EarlyMintAudit

Prepared By:

Delivery Date: May 20th, 2023

Auditor: Nickwang

Findings

High

[H-01] Users can bypass the ordersPerWallet restriction

Affected Functions

- `EarlyMint`
  - `function reserveOrder( uint256 _campaignId, uint8 _requestedOrderQuantity, bytes memory _signature )`

Description

msg.sender can break through this restriction `require (paidOrders[_campaignId] [msg.sender] + _ requestedOrderQuantity <= campaign.ordersPerWallet)` by calling `updateWalletAddress (uint256 _campaignId, address newAddress)`

calling sequence: `reserveOrder -> updateWalletAddress -> reserveOrder`

Mitigation

Determine the return value in `addPaidOrdersAddress(_campaignId, msg.sender)`

```
function addPaidOrdersAddress(
    uint256 _campaignId,
    address _paidOrdersAddress
) internal {
    bool exists = false;
    for (uint i = 0; i < paidOrdersAddresses[_campaignId].length; i++
        if (paidOrdersAddresses[_campaignId][i] == _paidOrdersAddress
            exists = true;
            break;
        }
    }

    if (!exists) {
        paidOrdersAddresses[_campaignId].push(_paidOrdersAddress);
    } else {
        return;
    }
}
```

Medium

[M-01] When creating a campaign, there was no verification of fee==0

Affected Functions

- `EarlyMint`
  - `executeMintForCampaign(uint256 _campaignId)`

Description

If `fee==0`, this agreement cannot receive fee

Mitigation

when call `createCampaign(Campaign memory _campaign, bytes memory _signature)`, add a limit on the number of fee in `_campaign.fee`, for example `_campaign >= FEE`

[M-02] Users may not be able to withdraw their funds

Affected Functions

- `EarlyMint`
  - `requestRefund(uint256 _campaignId)`

Description

If the user first calls `updateWalletAddress (uint256 _campaignId, address newAddress)` and then call `requestRefund (uint256 _campaignId)`, the user cannot withdraw their funds, because `paidOrders[_campaignId][msg.sender] == 0`

Mitigation

[M-03] No parameter validation was performed on newAddress

Affected Functions

- `EarlyMint`
  - `updateWalletAddress(uint256 _campaignId, address _newAddress)`

Description

No parameter validation was performed on `newAddress`, if call `updateWalletAddress` when `newAddress == 0`, function `earlyMint` in contract will be revert

Mitigation

```
require(_newAddress != 0x0)
```

Low

[L-01] updateWalletAddress can only be called once

Affected Functions

- `EarlyMint`
  - `updateWalletAddress(uint256 _campaignId, address _newAddress)`

Description

if `msg.sender != newAddress`,
because of `paidOrders[_campaignId][msg.sender] == 0`,
this function can be revert

[L-02] function checkWhitelistOrdersForAddress is not implemented in EarlyMintInterface

Affected Functions

- `EarlyMintTestERC721A`
  - `preMint(uint256 _mintAmount)`

Description

function `checkWhitelistOrdersForAddres` s is not implemented in `EarlyMintInterface`, when call function `preMint()`, it will be revert

[L-03] in modifier ensureUniqueCampaignExternalId, There may be gas exceeding the transaction limit, or users spending more gas in a transaction

Affected Functions

- `EarlyMint`
  - `createCampaign(Campaign memory _campaign, bytes memory _signature)`

Description

in modifier `ensureUniqueCampaignExternalId`, There may be gas exceeding the transaction limit, or users spending more gas in a transaction

Mitigation

Change for loop to `map(uint256=>bool) isExist` to find externalId is exist;
`requrie (!isExist[externalId])`