ERC721M

Findings and Recommendations Report Presented to:

Magic Eden

September 28, 2022
Version: 1.0

Presented by:

Kudelski Security, Inc.
5090 North 40th Street, Suite 450
Phoenix, Arizona 85018

For Public Release

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Overview

Magic Eden engaged Kudelski Security to perform a secure code assessment of the ERC721M launchpad contract.

The assessment was conducted remotely by the Kudelski Security Team. Testing took place on Sept 14, 2022 – Sept 28, 2022, and focused on the following objectives:

- Provide the customer with an assessment of the security of the ERC721M launchpad contract
- To provide a professional opinion on the maturity, efficiency, and coding practices
- To identify potential issues and include improvement recommendations based on the result of our tests.

This report summarizes the engagement, tests performed, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the Kudelski Security Teams took to identify and validate each issue, as well as any applicable recommendations for remediation.

Key Findings

During the test, the following positive observations were noted regarding the scope of the engagement:

- Test coverage was excellent
- Nice use of open-source libraries to augment functionality
- The introduction of stages allows for a wide range of use-cases

Scope and Rules Of Engagement

Kudelski performed a full assessment of ERC721M launchpad contract for Magic Eden. The following table documents the targets in scope for the engagement. No additional systems or resources were in scope for this assessment.

In-Scope Contracts	
magiceden-oss/erc721m	Commit: e5d675e60973107dcef6d410a04b0848cbdebf35

Table 1: Scope

TECHNICAL ANALYSIS & FINDINGS

During the ERC721M, we discovered 1 finding that had an informational severity rating.

The following chart displays the findings by severity.

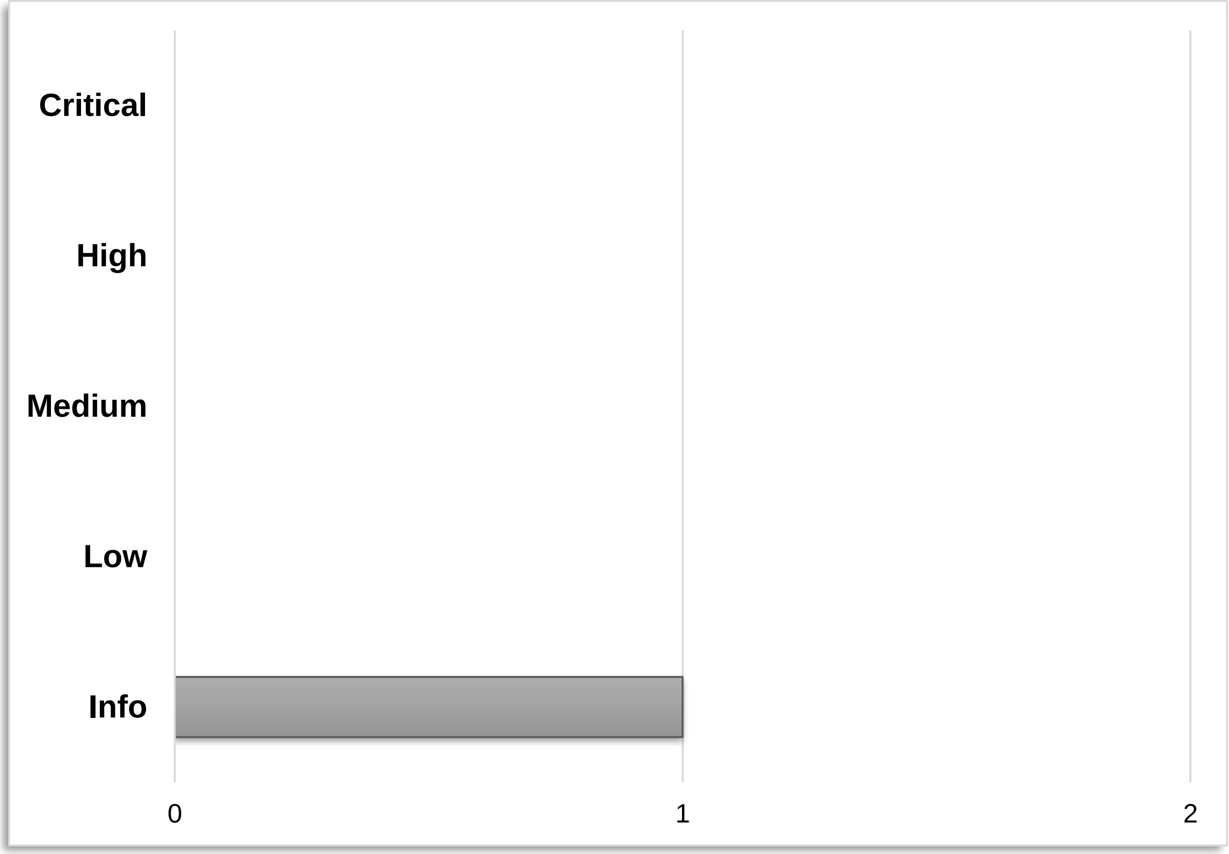


Figure 1: Findings by Severity

Findings

The *Findings* section provides detailed information on each of the findings, including methods of discovery, explanation of severity determination, recommendations, and applicable references.

The following table provides an overview of the findings.

#	Severity	Description
1	**Informational**	Owner may change stages at any time

Table 2: Findings Overview

1 – Stages are controlled by the contract owner

Severity	Informational

Impact	Likelihood	Difficulty
N/A	N/A	N/A

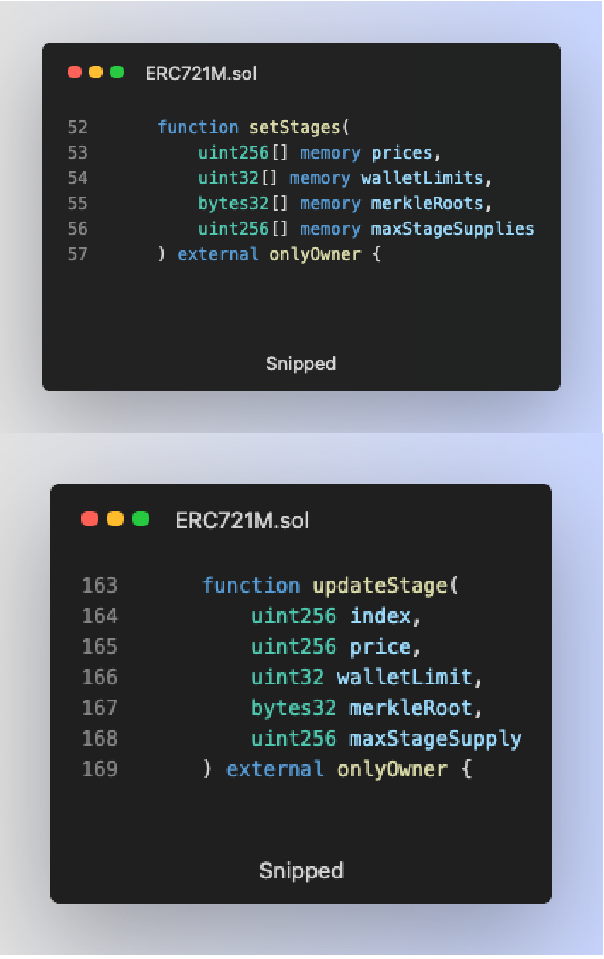
Description

ERC721M gave the contract owner full control over the minting stages. This includes ability 1) to change back to an earlier stage, and 2) to change a stage as the mint process is happening.

Impact

Users may view some of the features as unexpected. There is a risk of upset users if they are not aware of exactly what could happen during the minting process.

Evidence



ERC721M.sol

```
function setStages(
    uint256[] memory prices,
    uint32[] memory walletLimits,
    bytes32[] memory merkleRoots,
    uint256[] memory maxStageSupplies
) external onlyOwner {
```

Snipped

ERC721M.sol

```
function updateStage(
    uint256 index,
    uint256 price,
    uint32 walletLimit,
    bytes32 merkleRoot,
    uint256 maxStageSupply
) external onlyOwner {
```

Snipped

The contract owner has permission to set or update stages to anything at any time

Affected Resource

- ERC721M.sol

Recommendation

Magic Eden should make it clear to users that the project owners are in control of the minting stages, and that the minting stages could be changed to anything by project owners.

METHODOLOGY

During this source code review, the Kudelski Security Services team reviewed code within the project within an appropriate IDE. During every review, the team spends considerable time working with the client to determine correct and expected functionality, business logic, and content to ensure that findings incorporate this business logic into each description and impact. Following this discovery phase the team works through the following categories:

- Authentication
- Authorization and Access Control
- Configuration Issues
- Logic Flaws
- Cryptography

Tools

The following tools were used during this portion of the test.

- Visual Studio Code with Solidity extension
- Slither

Vulnerability Scoring Systems

Kudelski Security utilizes a vulnerability scoring system based on impact of the vulnerability, likelihood of an attack against the vulnerability, and the difficulty of executing an attack against the vulnerability based on a high, medium, and low rating system

Impact
The overall effect of the vulnerability against the system or organization based on the areas of concern or affected components discussed with the client during the scoping of the engagement.

High:
The vulnerability has a severe effect on the company and systems or has an effect within one of the primary areas of concern noted by the client

Medium:
It is reasonable to assume that the vulnerability would have a measurable effect on the company and systems that may cause minor financial or reputational damage.

Low:
There is little to no effect from the vulnerability being compromised. These vulnerabilities could lead to complex attacks or create footholds used in more severe attacks.

Likelihood
The likelihood of an attacker discovering a vulnerability, exploiting it, and obtaining a foothold varies based on a variety of factors including compensating controls, location of the application, availability of commonly used exploits, and institutional knowledge

High:
It is extremely likely that this vulnerability will be discovered and abused

Medium:
It is likely that this vulnerability will be discovered and abused by a skilled attacker

Low:
It is unlikely that this vulnerability will be discovered or abused when discovered.

Difficulty
Difficulty is measured according to the ease of exploit by an attacker based on availability of readily available exploits, knowledge of the system, and complexity of attack. It should be noted that a LOW difficulty results in a HIGHER severity.

Low:
The vulnerability is easy to exploit or has readily available techniques for exploit

Medium:
The vulnerability is partially defended against, difficult to exploit, or requires a skilled attacker to exploit.

High:
The vulnerability is difficult to exploit and requires advanced knowledge from a skilled attacker to write an exploit

Severity
Severity is the overall score of the weakness or vulnerability as it is measured from Impact, Likelihood, and Difficulty