SOLIDITY AUDIT

LANDMINING NFT

Contents

Project Information

Project Name: LandminingNFT
Network: Polygon (Polygonscan)
Contract Address:
0xE34553ec5c226432ea12e6A86734676A48E56c50
Type: NFT Staking
Max amount: N/A
Min amount: 5 MATIC
Dev fee: 10%
Owner fee: 0%
Marketing: 0%
Withdraw system:
Anytime
Minimum Withdrawal: 0 MATIC
Referral: 7%, 3%, 1%
Withdraw referral: Automatic
Max Referral: Unlimited
Reward system: Daily from 2% - 2.5%

Scope of Audit

The scope of this audit was to analyze and document the landminingNFT smart contract codebase for quality, security, and correctness. Any errors or incorrect will be fixed by our team.

Check Vulnerabilities

We scanned and checked those commonly known and specific vulnerabilities based on the solidity code of the project. Here are some of the commonly known vulnerabilities that we considered.

- Compiler
- Struct
- Plan
- Functions
- Users Checkpoint
- Timestamp
- Gas Limit and Loops
- Byte array
- Backdoor
- Transfer forward
- Redundant fallback

Number of issues per severity

Type	High	Medium	Low	Informational
Open	0	0	0	0
Acknowledge	0	0	2	1
Closed	0	0	2	1

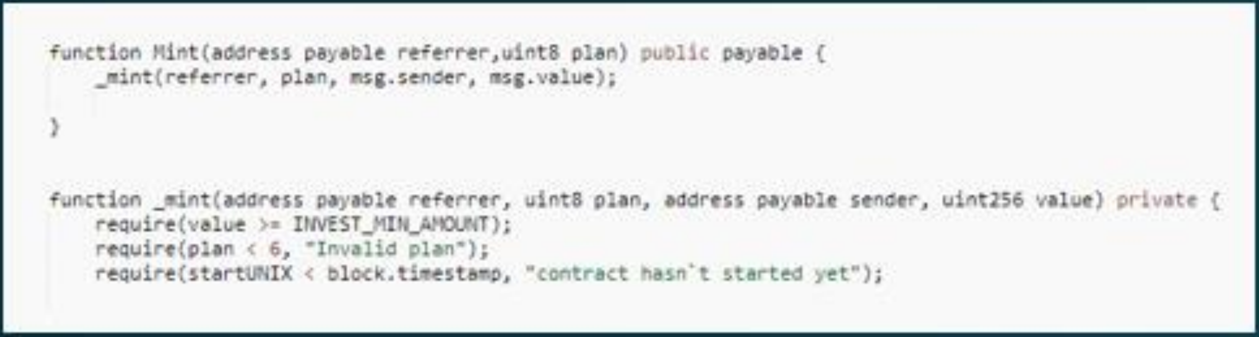
Transaction Code

There are few transactions code base of the smart contract of this project. Each transaction will be written on the blockchain, and no one can be manipulated or change.

1. **LandminingNFT (ROI)**

   1. **Launch Function**
   2. **Withdraw Function**
   3. **Mint Function**

2. Mint

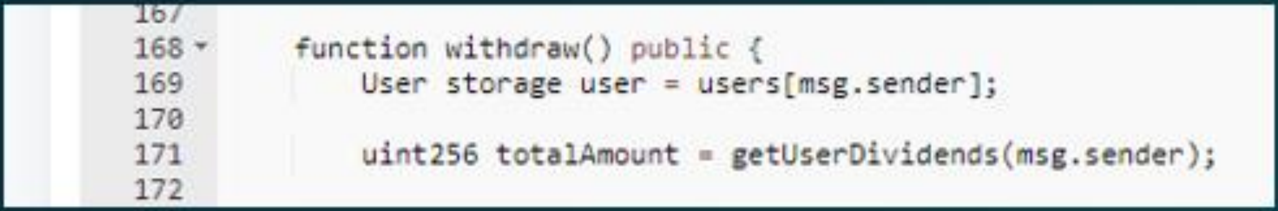


```
function Mint(address payable referrer,uint8 plan) public payable {
    _mint(referrer, plan, msg.sender, msg.value);

}


function _mint(address payable referrer, uint8 plan, address payable sender, uint256 value) private {
    require(value >= INVEST_MIN_AMOUNT);
    require(plan < 6, "Invalid plan");
    require(startUNIX < block.timestamp, "contract hasn`t started yet");
```

3. Withdraw



```

    function withdraw() public {
        User storage user = users[msg.sender];

        uint256 totalAmount = getUserDividends(msg.sender);

```

Overall function transaction will be seen at write contract overview. No backdoor or further function can cause of draining the contract.

1. **Launch**
2. **Mint**
3. **Withdraw**

Compiler Test

Compiled and tested under 3rd party solidity compiler.

0.5.8+commit.23d335f2	SUCCESS
No Warning Detected	SUCCESS
Compiled Success	SUCCESS

Contract Testing

Contract deployed and tested with our team before we released the audited report.

Network:

Polygon Testnet Mumbai

Deployed:

08.08.22

Contract Link:

https://mumbai.polygonscan.com/address/0xdECaf0ECc0a107aF84422Aa8F0807e7F302B2CE6

Result

No major issue was found. Some false positive errors were reported by the tool. All the other issues have been categorized according to their level of severity, the team improved and fixed it before the publicity of the project.

Closing Summary

Overall, smart contracts are very well written and adhered to guidelines. Minor issues were discovered during the audit that has been fixed.