Ascendant

VEGAS LUCK NFT READ-THROUGH

8/21/2022

SUMMARY:

I was requested to review the following contracts for vulnerabilities and rug pull mechanisms. This review contains comments and recommendations for the following contracts:

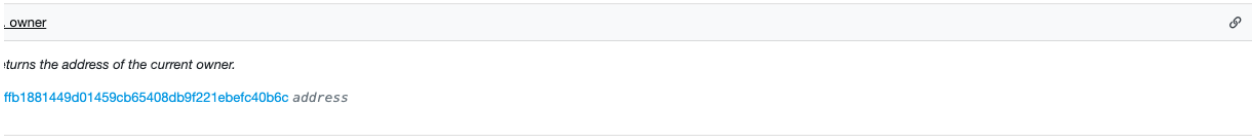
- VEGAS LUCK NFT: 0xa8F79C6aB5183e9429bD05667d3E0D77CAaeA225
- https://etherscan.io/address/0xa8F79C6aB5183e9429bD05667d3E0D77CAaeA225

COMMENTS:

Contract is **SAFE** and free of any rug pull mechanisms or mechanisms designed to siphon liquidity.

Ownership: Ownership has been transferred and the contract can no longer be controlled by anyone other than the current owner.

TX Receipt:
https://etherscan.io/tx/0xa07464e78d4d890749155ce1d776ae7c62336063c3690ccb534c23bf758bdbc7



Peripheral Contracts: 0x50b2c150bda3bef30373481c464a3a44215dc565 (DIV_DISTRIBUTOR)

Beneficiaries:

Payment Wallet: 0xffB1881449d01459cb65408Db9F221eBefc40b6C

ISSUES

ISSUE#1:

Vegas Luck NFT contract is the owner of DIV_DISTRIBUTOR, which renders the function withdrawfromDistributor() useless:

```
function withdrawFromDistributor() external onlyOwner {
        DIV_DISTRIBUTOR.withdrawETH(msg.sender);
```

```
    }
```

This is because when the withdrawETH function within the DIV_DISTRIBUTOR contract is called, the msg.sender in this instance is actually the contract, which is sending the balance back to itself.

Solution: Just use the withdraw function only, if necessary.

ISSUE#2

Contract uses block.timestamp to achieve pseudo-randomness:

```
function random(address _addr) private view returns(uint256){
        return uint256(keccak256(abi.encodePacked(block.difficulty, block.timestamp,
_addr))) % 10000000000;
    }
```

Using block.timestamp is dangerous when trying to generate random numbers because it can be manipulated and exploited by miners.

Solution: Consider off-chain randomization techniques, such as implementing a random number generator into the front-end instead.

ISSUE#3

Private address is published in the contract:

```
address private PAYMENT_WALLET = address(0xffB1881449d01459cb65408Db9F221eBefc40b6C)
```

This is only a major security issue if you intended to keep the payment wallet hidden. There's no real point to setting this address to private if it will be hardcoded into the contract.

Solution: If you intended to keep this address a secret, then you must redeploy, and instead set the payment wallet in the constructor. This way, it is not published. Otherwise, you can disregard this issue.

Kevan J. Williams, Esq.
Ascendant, LLC
https://ascendant.finance
thetechjd@gmail.com
Twitter: @ascendantproj