"DSQ Solutions Staking" Audit

Contract

Address : https://etherscan.io/address/0xb15dB3793ff2B0968aFe1A204064992991C53445

1: Libraries / Interfaces / Contract Inheritance: SafeMath / IERC165 / IERC721 / IERC20 / Context / Ownable / Nothing suspicious found in any of these although some library functions are not used in the main contract

2: Default variables Base APR = 5%, nft1BonusAPR = 1%, nft5BonusAPR = 2%, nft10BonusAPR = 3%, cooldownPeriod = 30 days, earlyWithdrawPenalty = 10%

lockperiods of 30 days with 0% bonus APR / 90 days with 1% bonus APR / 180 days with 2% bonus APR

3: Owner can turn staking on / off (not an issue in a staking setup)

4: Owner can remove stuck tokens from the contract but their are checks on dividendToken to prevent them from removing any of those while totalStaked > 0

5: Owner can remove any ETH held in the contract to their wallet (not a security concern with staking)

6: Owner can reset staking for any wallet (this sends the tokens back to the wallet in question and is not a security concern as such)

7: Owner can reset user cooldown setting their cooldown to 0 (not a security concern)

8: Owner can set a cooldown period of up to 90 days on any wallet that performs an emergency withdrawal (not a security concern)

In terms of security the contract is SAFE.

Marshmallow Man Audits

DAPP

Main App

1: Imports stakingABI and tokenABI from './abis.js' EthereumClient, w3mConnectors, w3mProvider from '@web3modal/ethereum' Web3Modal from '@web3modal/html' configureChains, createConfig, writeContract, readContract, waitForTransaction, watchAccount, fetchBalance from '@wagmi/core' Web3 from "web3"; mainnet from '@wagmi/core/chains' LineChart from './components/charts.vue' Moralis from 'moralis';

Nothing odd found in any import being used by the DAPP

2: Relevant constant variables tokenAddr = '0x7340ea46360576dc46ef49bce99bc5072c32421d' const NFTAddr = '0x9b6317A42133E9f247e967C0B7F34fFBd717d1Da' const StakingAddr = '0xb15dB3793ff2B0968aFe1A204064992991C53445'

All point to correct addresses for each contract

3: Notable functions makeApiCall() / maxAmount() / stakeMoreTokens() / deposit() / getuserInfo() / firstCheck() / calculatorModal() / apymodal() / calculateTokens() / harvest() / compound() / withdraw()

All of these are either calling relevant contract functions by use of the variables listed above for address or simply performing calculations to display, given all contracts have been audited above and prior I see nothing unsafe.

4: Pretty much everything else is a visual element for the DAPP itself.

Marshmallow Man Audits

Summary

The contracts being targeted are the DSQ token, staking and NFT contracts and the functions being used are listed above, none are dangerous. Keep in mind a DAPP can only display and use features built into contract code which is part of this audit and details are above, I see nothing that points to a security flaw in the staking DAPP combined with the above staking contract audit and previous token / NFT contract audits.