Coinsult

Advanced Manual Smart Contract Audit

Project: Beatn
Website: https://beatn.app/

Low-risk	Medium-risk	High-risk
3 low-risk code issues found	0 medium-risk code issues found	0 high-risk code issues found

Contract address
0x520B1e2258b95B0DdaAe0d09af353ffA66616268

Disclaimer

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xf7a2d448df3f80196ccb5f5dc90a0e68a080f5dc	1,000,000	100.000%

Source code

Coinsult was commissioned by Beatn to perform an audit based on the following smart contract:

https://bscscan.com/address/0x520b1e2258b95b0ddaae0d09af353ffa66616268#code

Manual Code Review

🟢 Low-risk

3 low-risk code issues found.
Could be fixed, will not bring problems.

- Contract contains Reentrancy vulnerabilities:
  Additional information: This combination increases risk of malicious intent. While it may be justified by some complex mechanics (e.g. rebase, reflections, buyback).
  More information: Slither

```
    function _transferFrom(
        address sender,
        address recipient,
        uint256 amount
    ) internal returns (bool) {
        // antiBot
        if ((sender == pair) || (recipient == pair)) {
            if (!isAntiBotEnded()) {
                if (!(addLPaddress[sender] || addLPaddress[recipient]))
{
                    require(amount <= antiBotAmount, "antiBot");
                }
            }
        }
        // antiBot

        if (inSwap) {
            return _basicTransfer(sender, recipient, amount);
        }
        if (shouldRebase()) {
            rebase();
        }
        if (shouldAddLiquidity()) {
            addLiquidity();
        }
        if (shouldSwapBack()) {
            swapBack();
        }

        uint256 gonAmount = amount.mul(_gonsPerFragment);
```

```
        _gonBalances[sender] = _gonBalances[sender].sub(gonAmount);
        uint256 gonAmountReceived = shouldTakeFee(sender, recipient)
            ? takeFee(sender, recipient, gonAmount)
            : gonAmount;
        _gonBalances[recipient] = _gonBalances[recipient].add(
            gonAmountReceived
        );

        emit Transfer(
            sender,
            recipient,
            gonAmountReceived.div(_gonsPerFragment)
        );
        return true;
    }
```

- Avoid relying on block.timestamp
  block.timestamp can be manipulated by miners.

```
function setAntiBotAmount(uint256 _amount) public onlyOwner {
        antiBotAmount = _amount;
        antiBotStart = block.timestamp - 10; // safety margin 10s
    }
```

- Missing zero address validation
  Check that the new address is not the zero address.

```
       pairContract = IPancakeSwapPair(_address);
```

🟡 Medium-risk

0 medium-risk code issues found.
Should be fixed, could bring problems.

🔴 High-risk

0 high-risk code issues found
Must be fixed, and will bring problems.

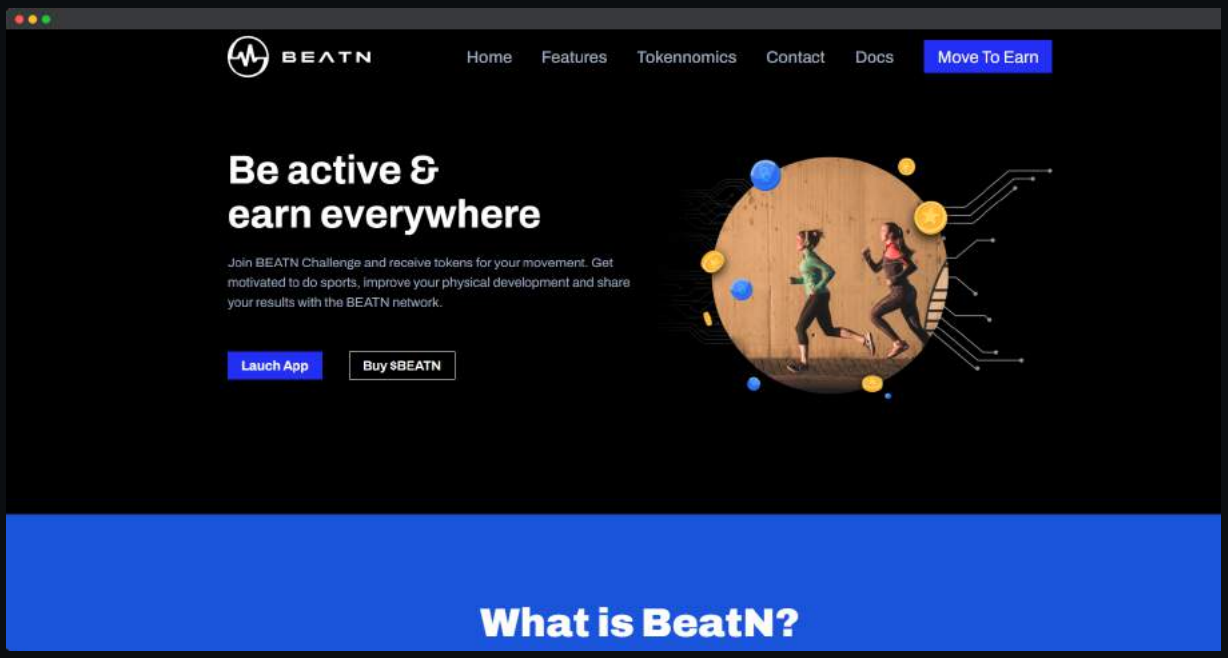
Extra notes by the team

- 🟢 Use of anti bot mechanism
- 🟢 Fees are hardcoded
- 🟡 The ownership of the contract isn’t renounced.
- 🟡 Owner can change the router address
- 🟡 Owner can exclude addresses from fees.

Contract Snapshot

```
contract BEATN is ERC20Detailed, Ownable {
    using SafeMath for uint256;
    using SafeMathInt for int256;

    event LogRebase(uint256 indexed epoch, uint256 totalSupply);

    string public constant _name = "BEATN";
    string public constant _symbol = "BEATN";
    uint8 public constant _decimals = 5;

    IPancakeSwapPair public pairContract;
    mapping(address => bool) _isFeeExempt;

    modifier validRecipient(address to) {
        require(to != address(0x0));
        _;
    }

    uint256 public constant DECIMALS = 5;
    uint256 public constant MAX_UINT256 = ~uint256(0);
    uint8 public constant RATE_DECIMALS = 7;

    uint256 private constant INITIAL_FRAGMENTS_SUPPLY =
        100 * 10**4 * 10**DECIMALS;

    uint256 public constant liquidityFee = 40;
    uint256 public constant treasuryFee = 20;
    uint256 public constant beatnRiskFreeValueFee = 40;
    uint256 public constant beatnNftHolderFee = 40;
    uint256 public constant firePitFee = 20;
```

Website Review



Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.

- Mobile Friendly
- Contains no jQuery errors
- SSL Secured
- No major spelling errors

Note: website is not yet fully operational. There are multiple buttons which do not work yet. For example the top-bar menu.

Loading speed: 72%

Rug-pull Review

Based on the available information analyzed by us, we come to the following conclusions:

🟢 Locked Liquidity (no liquidity yet)

🟡 Large unlocked wallets

- Note: Tokens not distributed yet

🔴 No doxxed Team

Honeypot Review

Based on the available information analyzed by us, we come to the following conclusions:

🟡 Ability to sell

- Note: Owner can set antibot amount really low, which in theory, could prevent any selling taking place.

🟢 Owner is not able to pause the contract

🟡 Router can be changed

Note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.