Coinsult

Advanced Manual Smart Contract Audit

Project: Bloggercube

Website: https://bloggercube.com

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

Contract Address

0x43B121Af0fE2085D72c544e4B6f163C5A8a15D9F

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	0x8dbef490ed14e0e54747a53f64f0def6a8384bb6	6,136,200,000	61.3620%
2	0xd8c979a29ac543a598dc6b94b901bb0b18d66a1b	3,263,800,000	32.6380%
3	0xbacabac66a434645be09f8ff15c3f05555279c88	200,000,000	2.0000%
4	0x60f5ae54a9c1fe3d310cf01351c4dff96e92fe02	200,000,000	2.0000%
5	0x57a178d61caf61bec3364286c196b8dc54f7e43e	200,000,000	2.0000%

Source Code

Coinsult was comissioned by Bloggercube to perform an audit based on the following smart contract:

https://bscscan.com/address/0x43B121Af0fE2085D72c544e4B6f163C5A8a15D9F#code

Manual Code Review

In this audit report we will highlight all these issues:

Low-Risk

5 low-risk code
issues found

Medium-Risk

0 medium-risk code
issues found

High-Risk

0 high-risk code
issues found

The detailed report continues on the next page...

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Too many digits

Literals with many digits are difficult to read and review.

```
_mint(msg.sender, 10000000000 * 10**decimals());
```

Recommendation

Use: Ether suffix, Time suffix, or The scientific notation

Exploit scenario

```
contract MyContract{
    uint 1_ether = 10000000000000000000;
}
```

While `1_ether` looks like `1 ether`, it is `10 ether`. As a result, it's likely to be used incorrectly.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

No zero address validation for some functions

Detect missing zero address validation.

```
function setMarketingWalletAddress(address _addr)
  public
  onlyOwner
  returns (bool)
{
  marketingWalletAddress = _addr;
  return true;
}
```

Recommendation

Check that the new address is not zero.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

Bob calls `updateOwner` without specifying the `newOwner`, soBob loses ownership of the contract.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Divide before multiply

Solidity integer division might truncate. As a result, performing multiplication before division can sometimes avoid loss of precision.

```
if (marketingWalletAddress != address(0) &amp;amp;&amp;amp; marketingFeePercent &amp;gt; 0) {
        marketingFee = (amount / 100) * marketingFeePercent;
      }
      if (developmentWalletAddress != address(0) &amp;amp;&amp;amp; developmentFeePercent &amp;gt; 0) {
        developmentFee = (amount / 100) * developmentFeePercent;
      }
      if (donationWalletAddress != address(0) &amp;amp;&amp;amp; donationFeePercent &amp;gt; 0) {
        donationFee = (amount / 100) * donationFeePercent;
      }
```

Recommendation

Consider ordering multiplication before division.

Exploit scenario

```
contract A {
    function f(uint n) public {
        coins = (oldSupply / n) * interest;
    }
}
```

If `n` is greater than `oldSupply`, `coins` will be zero. For example, with `oldSupply = 5; n = 10, interest = 2`, coins will be zero. If `(oldSupply * interest / n)` was used, `coins` would have been `1`. In general, it's usually a good idea to re-arrange arithmetic to perform multiplication before division, unless the limit of a smaller type makes this dangerous.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
function setMarketingFee(uint8 _fee)
  public
  onlyOwner
  onlyValidFee(_fee)
  returns (bool)
{
  marketingFeePercent = _fee;
  return true;
}
```

Recommendation

Emit an event for critical parameter changes.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

`updateOwner()` has no event, so it is difficult to track off-chain changes in the buy price.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Conformance to Solidity naming conventions

Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
donationWalletAddress = _addr
```

Recommendation

Follow the Solidity naming convention.

Rule exceptions

- Allow constant variable name/symbol/decimals to be lowercase (`ERC20`).
- Allow _ at the beginning of the `mixed_case` match for private variables and unused parameters.

Owner privileges

- Owner cannot pause trading
- Owner cannot change max transaction amount
- Owner cannot set fees higher than 30%

Extra notes by the team

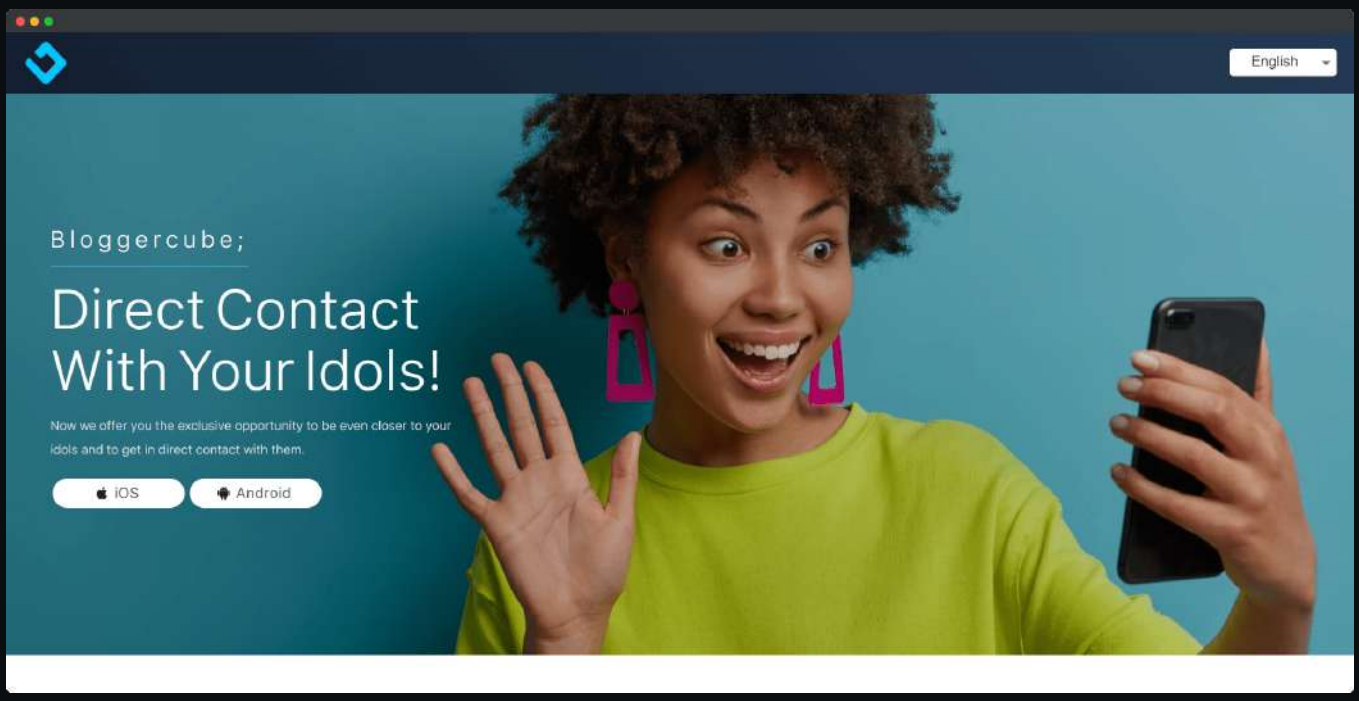
No notes

Contract Snapshot

```
contract BloggercubeToken is ERC20, ERC20Snapshot, Ownable {
// State Variables
uint8 public marketingFeePercent = 0;
uint8 public developmentFeePercent = 0;
uint8 public donationFeePercent = 0;
address public marketingWalletAddress;
address public developmentWalletAddress;
address public donationWalletAddress;
address public _owner;
```

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
- Does not contain jQuery errors
- SSL Secured
- No major spelling errors

Project Overview

Not KYC verified by Coinsult