Coinsult

Advanced Manual Smart Contract Audit

Project: BSCFlip (dapp contract)
Website: https://www.bscflip.com

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

5 acknowledged & fixed

Contract address
0x601Cf418268900e4E75E826B69C649367e1028e3 (redeploy)

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

Inapplicable

Source code

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

https://bscscan.com/address/0x601Cf418268900e4E75E826B69C649367e1028e3#code (redeployed)

Manual Code Review

🟢 Low-risk

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

- No zero address validation

  Check that the new address is not the zero address

```
    function setDevFeeReceiver(address newReceiver) external onlyOwner
{
        require(newReceiver != _devFeeReceiver, "This is already the
dev fee receiver");

        _devFeeReceiver = newReceiver;
    }
```

🟢 Acknowledged and fixed:

```
  function setDevFeeReceiver(address newReceiver) external onlyOwner {
      require(newReceiver != address(0x0), "Can't set the zero address
as the receiver");
      require(newReceiver != _devFeeReceiver, "This is already the dev
fee receiver");

      emit DevFeeReceiverChanged(_devFeeReceiver, newReceiver);

      _devFeeReceiver = newReceiver;
  }
```

- No zero address validation

  Check that the new address is not the zero address

```
    function setHouseFeeReceiver(address newReceiver) external
onlyOwner {
        require(newReceiver != _houseFeeReceiver, "This is already the
house fee receiver");

        _houseFeeReceiver = newReceiver;
    }
```

🟢 Acknowledged and fixed:

```
  function setHouseFeeReceiver(address newReceiver) external onlyOwner
{
      require(newReceiver != address(0x0), "Can't set the zero address
as the receiver");
      require(newReceiver != _houseFeeReceiver, "This is already the
house fee receiver");

      emit HouseFeeReceiverChanged(_houseFeeReceiver, newReceiver);

      _houseFeeReceiver = newReceiver;
  }
```

- `SafeMath` is generally not needed starting with Solidity 0.8, since the compiler now has built-in overflow checking.

```
library SafeMath {
```

- Contract contains a lot of commented code, which can be removed

🟢 Acknowledged and fixed

```
    /**
     * @dev Tool to verifies that a low level call was successful, and
revert if it wasn't, either by bubbling the
     * revert reason using the provided one.
     *
     * _Available since v4.3._
     */
```

- Emit an event for critical parameter changes.

```
    function setHouseFeePercentage(uint8 newPercentage) external
onlyOwner {
        require(newPercentage != _houseFeePercentage, "This is already
the house fee percentage");
        require(newPercentage <= 30, "Cannot set house fee percentage
higher than 3 percent");

        _houseFeePercentage = newPercentage;
    }
```

🟢 Acknowledged and fixed:

```
  function setHouseFeePercentage(uint8 newPercentage) external
onlyOwner {
      require(newPercentage != _houseFeePercentage, "This is already
the house fee percentage");
      require(newPercentage <= 40, "Cannot set house fee percentage
higher than 4 percent");

      emit HouseFeePercentageChanged(_houseFeePercentage,
newPercentage);

      _houseFeePercentage = newPercentage;
  }
```

- Emit an event for critical parameter changes.

```
    function setDevFeePercentage(uint8 newPercentage) external
onlyOwner {
        require(newPercentage != _devFeePercentage, "This is already
the dev fee percentage");
        require(newPercentage <= 5, "Cannot set dev fee percentage
higher than 0.5 percent");

        _devFeePercentage = newPercentage;
    }
```

🟢 Acknowledged and fixed:

```
  function setDevFeePercentage(uint8 newPercentage) external onlyOwner
{
      require(newPercentage != _devFeePercentage, "This is already the
dev fee percentage");
      require(newPercentage <= 5, "Cannot set dev fee percentage higher
than 0.5 percent");

      emit DevFeePercentageChanged(_devFeePercentage, newPercentage);

      _devFeePercentage = newPercentage;
  }
```

🟡 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

🟡 Owner can blacklist players

```
    function setBlacklist(address wallet, bool isBlacklisted) external
onlyTeam {
        _isBlacklisted[wallet] = isBlacklisted;
    }
```

🟡 Owner can set max and min bet without a limit

```
    function setMinBetForToken(address token, uint256 minBet) external
onlyTeam {
        _minBetForToken[token] = minBet;
    }

    function setMaxBetForToken(address token, uint256 maxBet) external
onlyTeam {
        _maxBetForToken[token] = maxBet;
    }
```

🟡 Contract could potentially have too few tokens to distribute winnings

```
            require(winnings <= gameToken.balanceOf(address(this)),
"Not enough tokens in the contract to distribute winnings");
```

🟡 Owner can pause the game

```
    function setGameEnabled(bool enabled) external onlyTeam {
        require(enabled != _gameEnabled, "Must set a new value for
gameEnabled");

        _gameEnabled = enabled;
    }
```