Coinsult

Advanced Manual Smart Contract Audit

Project: Ethereum Aqua SWAP

Website: https://aquaswap.net/

Low-Risk	Medium-Risk	High-Risk
5 low-risk code issues found	1 medium-risk code issues found	0 high-risk code issues found

Contract Address

0x2015a0c83076f68d76bEd6Af89C38DE2c1c7D650

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	0x57b1ecfb40db8a661a372e23c947b3c04c61fdf7	650,300,500	65.0301%
2	0x2015a0c83076f68d76bed6af89c38de2c1c7d650	349,009,000	34.9009%
3	0x943535da485e32f444377c17b9e112f356f68178	101,500	0.0102%
4	0xc79f5228d21dafd1e0fe5f1e87044835319e6ad7	64,000	0.0064%
5	0x63064478972346dfc086ebf9066eccbc6ff0ebe6	60,000	0.0060%

Source Code

Coinsult was comissioned by Ethereum Aqua SWAP to perform an audit based on the following smart contract:

https://bscscan.com/address/0x2015a0c83076f68d76bEd6Af89C38DE2c1c7D650#code

Token sniffer reports:
This token was flagged due to evidence of a bug, hack, or scam:
– Exploit #034: Fake ownership renounce

Check the medium risk issue for more details.

Manual Code Review

In this audit report we will highlight all these issues:

Low-Risk

5 low-risk code
issues found

Medium-Risk

1 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.

```
_totalSupply =  1000000000*10** uint(decimals);
balances[owner] = 650000000*10** uint(decimals);
balances[address(this)] = 350000000*10** uint(decimals);
emit Transfer(address(0), owner, 650000000*10** uint(decimals));
emit Transfer(address(0), address(this), 350000000*10** uint(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 transferOwnership(address _newOwner) public onlyOwner {
  newOwner = _newOwner;
}
```

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.

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
function startAirdrop(uint256 _aSBlock, uint256 _aEBlock, uint256 _aAmt, uint256 _aCap) public onlyO
  aSBlock = _aSBlock;
  aEBlock = _aEBlock;
  aAmt = _aAmt;
  aCap = _aCap;
  aTot = 0;
}
function startSale(uint256 _sSBlock, uint256 _sEBlock, uint256 _sChunk, uint256 _sPrice, uint256 _sC
  sSBlock = _sSBlock;
  sEBlock = _sEBlock;
  sChunk = _sChunk;
  sPrice =_sPrice;
  sCap = _sCap;
  sTot = 0;
}
```

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.

Boolean equality

Detects the comparison to boolean constants.

```
require (_hasClaimed[ msg.sender] != true, 'You have already claimed!');
```

Recommendation

Remove the equality to the boolean constant.

Exploit scenario

```
contract A {
    function f(bool x) public {
        // ...
        if (x == true) { // bad!
           // ...
        }
        // ...
    }
}
```

Boolean constants can be used directly and do not need to be compare to `true` or `false`.

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

Redundant Statements

Detect the usage of redundant statements that have no effect.

```
balances[address(this)] = balances[address(this)].sub(_tkns / 1);
balances[_refer] = balances[_refer].add(_tkns / 1);

No need to divide by 1
```

Recommendation

Remove redundant statements if they congest code but offer no value.

Exploit scenario

```
contract RedundantStatementsContract {

    constructor() public {
        uint; // Elementary Type Name
        bool; // Elementary Type Name
        RedundantStatementsContract; // Identifier
    }

    function test() public returns (uint) {
        uint; // Elementary Type Name
        assert; // Identifier
        test; // Identifier
        return 777;
    }
}
```

Each commented line references types/identifiers, but performs no action with them, so no code will be generated for such statements and they can be removed.

🟡 **Medium-Risk:** Should be fixed, could bring problems.

Can’t rennounce ownership

```
function transferOwnership(address _newOwner) public onlyOwner {
  newOwner = _newOwner;
}
function acceptOwnership() public {
  require(msg.sender == newOwner);
  emit OwnershipTransferred(owner, newOwner);
  owner = newOwner;
  newOwner = address(0);
}
```

Recommendation

Using this configuration, the ownership can never be renoucned because you can’t accept the ownership with the zero address. This configuration will also trigger some warnings on automated scanning tools because of the ‘newOwner = address(0)” value.

Owner privileges

- Owner cannot set fees higher than 25%
- Owner cannot pause trading
- Owner cannot change max transaction amount

Extra notes by the team

No notes

Contract Snapshot

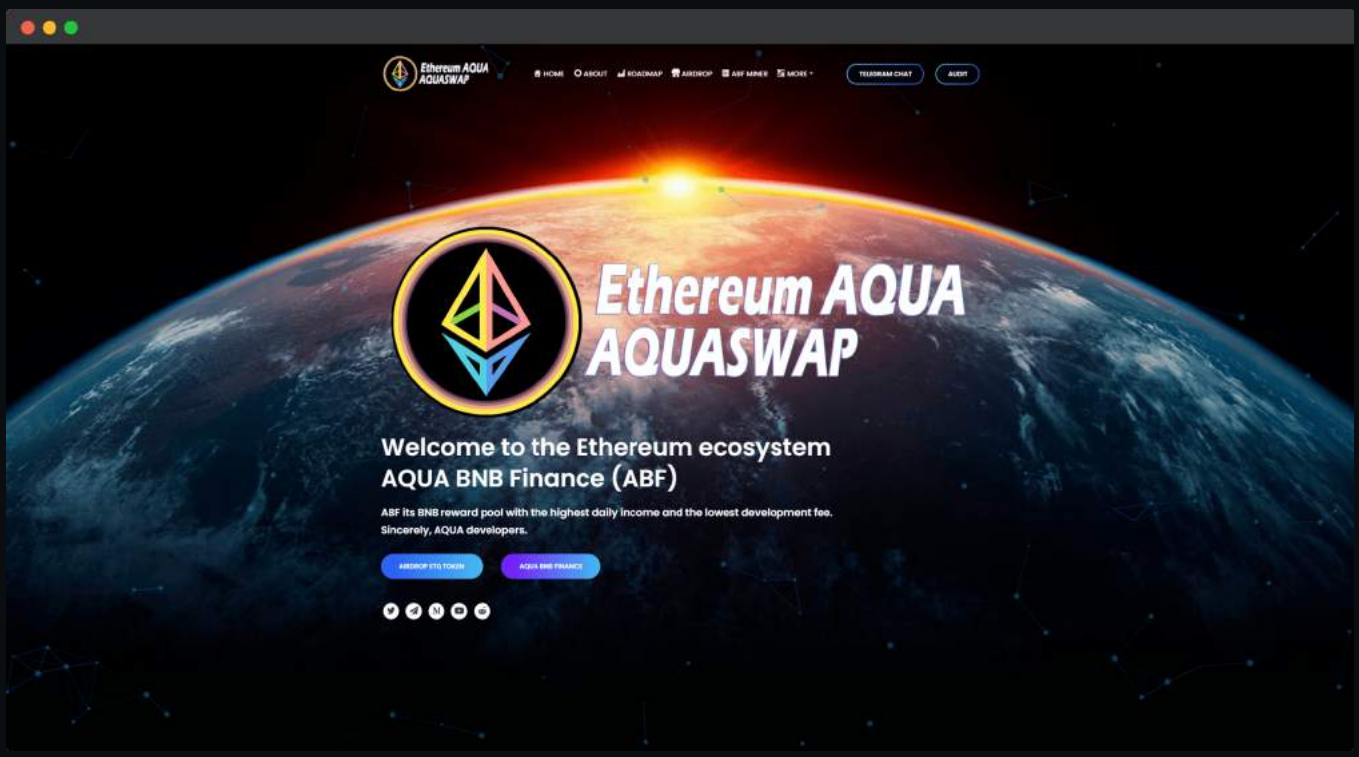
```
contract EthereumAquaSwap is TokenBEP20 {

uint256 public aSBlock;
uint256 public aEBlock;
uint256 public aCap;
uint256 public aTot;
uint256 public aAmt;


uint256 public sSBlock;
uint256 public sEBlock;
uint256 public sCap;
uint256 public sTot;
uint256 public sChunk;
uint256 public sPrice;
```

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