# Coinsult

# Advanced Manual Smart Contract Audit

**Project:** SpeedHero

**Website:** https://www.speedhero.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**

0xedCBDb4eAd2619Fe7e3D8b90A799c882a19c1054

# 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 | 0x8a3f5ca4c5ffa21ae65b81550e7045c6902137ad | 1,000,000,000,000,000 | 100.0000% |

# Source Code

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

https://bscscan.com/address/0xedcbdb4ead2619fe7e3d8b90a799c882a19c1054#code

# Manual Code Review

In this audit report we will highlight all these issues:

## 🟢 Low-Risk

3 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.

### Avoid relying on block.timestamp

block.timestamp can be manipulated by miners.

```
require(lastTrade[sender] &lt; (block.timestamp - tradeCooldown), string(&quot;No consecutive sells
```

### Recommendation

Do not use `block.timestamp`, now or `blockhash` as a source of randomness

### Exploit scenario

```
contract Game {

    uint reward_determining_number;

    function guessing() external{
      reward_determining_number = uint256(block.blockhash(10000)) % 10;
    }
}
```

Eve is a miner. Eve calls `guessing` and re-orders the block containing the transaction. As a result, Eve wins the game.

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

### No zero address validation for some functions

Detect missing zero address validation.

```
function setMarketingPool(address _marketingPool) external onlyOwner {
  marketingPool = _marketingPool;
  emit changeMarketingPool(_marketingPool);
}
```

### 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.

## Boolean equality

Detects the comparison to boolean constants.

```
if(liquidityPool[sender] == true) {      //It's an LP Pair and it's a buy

     taxAmount = (amount * buyTax) / 100;
} else if(liquidityPool[receiver] == true) {
     //It's an LP Pair and it's a sell
     taxAmount = (amount * sellTax) / 100;
```

## 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`.

## Owner privileges

- 🟢 Owner cannot set fees higher than 25%
- 🟢 Owner cannot pause trading
- 🟢 Owner cannot change max transaction amount
- 🟡 Owner can exclude from fees
- 🔴 Owner can blacklist addresses
- ⚠️ Owner can set cooldown between trades without a limit
- 🟢 Tax for buys, sells and transfers fee can be 10%

## Extra notes by the team

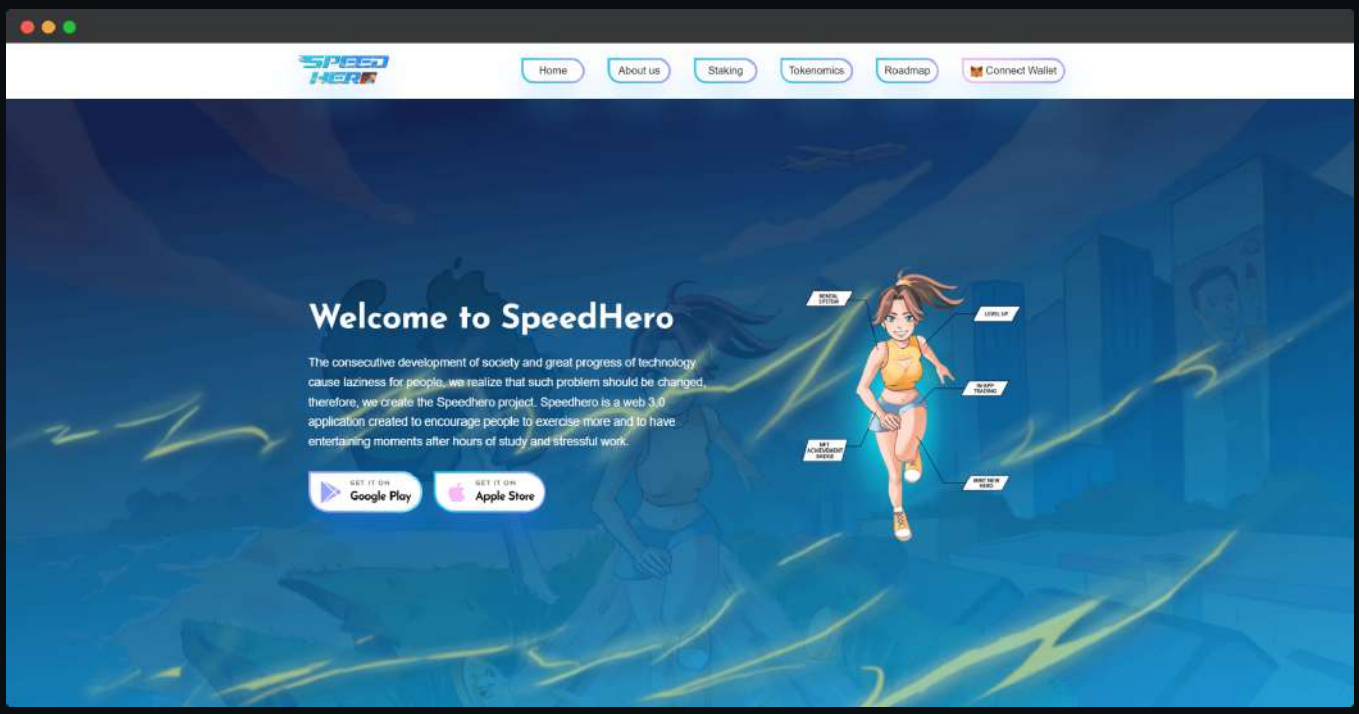
No notes

# Contract Snapshot

```
contract SpeedHero is BEP20Detailed, BEP20 {

mapping(address =&gt; bool) private isBlacklist;
mapping(address =&gt; bool) private liquidityPool;
mapping(address =&gt; bool) private whitelistTax;
mapping(address =&gt; uint256) private lastTrade;

uint8 private buyTax;
uint8 private sellTax;
uint8 private tradeCooldown;
uint8 private transferTax;
uint256 private taxAmount;
bool private lockSell = false;
bool private lockBuy = false;

address private marketingPool;

event changeBlacklist(address _wallet, bool status);
event changeCooldown(uint8 tradeCooldown);
event changeTax(uint8 _sellTax, uint8 _buyTax, uint8 _transferTax);
event changeLiquidityPoolStatus(address lpAddress, bool status);
event changeMarketingPool(address marketingPool);
event changeWhitelistTax(address _address, bool status);

constructor() BEP20Detailed("SpeedHero", "SHO", 18) {
  uint256 totalTokens = 1000 * 10 ** 12 * 10**uint256(decimals());
  _mint(msg.sender, totalTokens);
  sellTax = 3;
  buyTax = 0;
  transferTax = 0;
  tradeCooldown = 30;
  marketingPool = 0x46FCC7D4490E7d9416F50b2EbfeC5BA3d1BB5240;
}


function setBlacklist(address _wallet, bool _status) external onlyOwner {
  isBlacklist[_wallet]= _status;
```

## 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

KYC verified by Coinsult