



Coinsult

Advanced Manual Smart Contract Audit



Project: FTTS

Website: <https://www.ftts.me/>

Low-risk

4 low-risk code
issues found

Medium-risk

0 medium-risk code
issues found

High-risk

0 high-risk code
issues found

Contract address

0x669765d450e92d79fEBbB966a1eAD054cb62F33b

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

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 | 0xc3ffbdcdcde259f388da391aeafb1f396d4ae6cd9e | 21,000,000 | 100.000% |

Source code

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

<https://bscscan.com/address/0x669765d450e92d79febbb966a1ead054cb62f33b#code>

Manual Code Review

● Low-risk

4 low-risk code issues found.

Could be fixed, will not bring problems.

- Literals with many digits are difficult to read and review.
Recommendation: Use Ether suffix, Time suffix, or The scientific notation

```
_mint(_msgSender(), 21000000 * 10 ** decimals());
```

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

```
_fist, t1, t2, 0, 0, _main, block.timestamp);
```

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

```
function setAddr(address mark, address main) public {  
    require(_whites[_msgSender()]);  
    _mark = mark;  
    _main = main;  
}
```

- The return value of an external transfer/transferFrom call is not checked
Use SafeERC20, or ensure that the transfer/transferFrom return value is checked.

```
function _swapAndLiquid() private {  
    uint256 balances = balanceOf(address(this));  
    uint256 fistval = IERC20(_fist).balanceOf(address(this));  
    if (fistval > 0) {  
        IERC20(_fist).transfer(_mark, fistval);  
        return;  
    }  
  
    if (IERC20(_fist).allowance(address(this), _router) <= 10 ** 16  
        || allowance(address(this), _router) <= balances) {
```


● **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

- The ownership of the contract isn't renounced.
- Owner can set max transaction amount without constraints
- Fees can be set up to 100% for both buy and sell fees.
- Owner can exclude addresses from fees.
- Owner can change the router address

Contract Snapshot

```
contract Token is Ownable, IERC20Metadata {
    mapping(address => bool) public _whites;
    mapping(address => uint256) private _balances;
    mapping(address => mapping(address => uint256)) private
    _allowances;

    string private _name;
    string private _symbol;
    uint256 private _totalSupply;
    uint256 public _maxsell;

    uint256 public _burns;
    uint256 public _marks;
    uint256 public _pools;

    address public _router;
    address public _fist;
    address public _wrap;

    address public _main;
    address public _mark;
    address public _dead;

    constructor() {
        _name = "Fist Smart Tool";
        _symbol = "FTTS";

        _burns = 1;
        _pools = 2;
        _marks = 1;
        _maxsell = 5000 * 10 ** decimals();

        _main = 0x433DD4EC6a32846ff1fA4FD950DD451AAe482E5b;
        _mark = 0x252d1Afd05c384157522AA10fa93C969953dec66;
        _dead = 0x0000000000000000000000000000000000000000dEaD;

        _whites[_main] = true;
        _whites[_mark] = true;
        _whites[_msgSender()] = true;
        _whites[address(this)] = true;
    }
}
```

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 safe to enter.

Loading speed: 69%

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
- 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.