



# Coinsult

## Advanced Manual Smart Contract Audit



**Project:** Foreverise

**Website:** <https://foreverise.io/>

**Low-risk**

5 low-risk code  
issues found

**Medium-risk**

0 medium-risk code  
issues found

**High-risk**

0 high-risk code  
issues found

**Contract address**

0x3f2D74641779213de017eff7fc29F9420E797F84

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	0x25034aa02726370f535b593afeb0868da69c74ab	100,000,000,000	68.4953%

# Source code

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

<https://bscscan.com/address/0x3f2d74641779213de017eff7fc29f9420e797f84#code>

# Manual Code Review

## ● Low-risk

5 low-risk code issues found.

Could be fixed, will not bring problems.

- Contract contains Reentrancy vulnerabilities:

Additional information: This combination increases risk of malicious intent. While it may be justified by some complex mechanics (e.g. rebase, reflections, buyback).

More information: Slither

```
function _transferFrom(address sender, address recipient, uint256
amount) internal returns (bool) {
    if(inSwap){ return _basicTransfer(sender, recipient, amount); }

    if(!authorizations[sender] && !authorizations[recipient]){
        require(tradingOpen,"Trading not open yet");
    }

    // Blacklist
    if(blacklistMode){
        require(!isBlacklisted[sender] &&
!isBlacklisted[recipient],"Blacklisted");
    }

    if (!authorizations[sender] && recipient != address(this) &&
recipient != address(DEAD) && recipient != pair && recipient !=
marketingFeeReceiver && recipient != teamFeeReceiver && recipient !=
autoLiquidityReceiver && recipient != burnFeeReceiver){
        uint256 heldTokens = balanceOf(recipient);
        require((heldTokens + amount) <= _maxWalletToken,"Total Holding
is currently limited, you can not buy that much.");}

    if (sender == pair &&
        buyCooldownEnabled &&
        !isTimelockExempt[recipient]) {
        require(cooldownTimer[recipient] < block.timestamp,"Please wait
for 1min between two buys");
        cooldownTimer[recipient] = block.timestamp +
cooldownTimerInterval;
    }

    // Checks max transaction limit
```

```

        checkTxLimit(sender, amount);

        if(shouldSwapBack()){ swapBack(); }

        //Exchange tokens
        _balances[sender] = _balances[sender].sub(amount, "Insufficient
Balance");

        uint256 amountReceived = (!shouldTakeFee(sender) ||
!shouldTakeFee(recipient)) ? amount : takeFee(sender, amount, (recipient ==
pair));
        _balances[recipient] = _balances[recipient].add(amountReceived);

        // Dividend tracker
        if(!isDividendExempt[sender]) {
            try distributor.setShare(sender, _balances[sender]) {} catch {}
        }

        if(!isDividendExempt[recipient]) {
            try distributor.setShare(recipient, _balances[recipient]) {}
catch {}
        }

        try distributor.process(distributorGas) {} catch {}

        emit Transfer(sender, recipient, amountReceived);
        return true;
    }

```

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

```
block.timestamp
```

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

```

function setFeeReceivers(address _autoLiquidityReceiver, address
_marketingFeeReceiver, address _teamFeeReceiver, address
_burnFeeReceiver ) external authorized {
    autoLiquidityReceiver = _autoLiquidityReceiver;
    marketingFeeReceiver = _marketingFeeReceiver;
    teamFeeReceiver = _teamFeeReceiver;
    burnFeeReceiver = _burnFeeReceiver;
}

```

- Do not use write after write for the same variable usage

More information: Slither

```
tmpSuccess = false;

(bool tmpSuccess,) = payable(marketingFeeReceiver).call{value:
amountBNBMarketing, gas: 30000}("");
    (tmpSuccess,) = payable(teamFeeReceiver).call{value:
amountBNBTeam, gas: 30000}("");
```

- 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 distributeDividend(address shareholder) internal {
    if(shares[shareholder].amount == 0){ return; }

    uint256 amount = getUnpaidEarnings(shareholder);
    if(amount > 0){
        totalDistributed = totalDistributed.add(amount);
        RWRD.transfer(shareholder, amount);
        shareholderClaims[shareholder] = block.timestamp;
        shares[shareholder].totalRealised =
shares[shareholder].totalRealised.add(amount);
        shares[shareholder].totalExcluded =
getCumulativeDividends(shares[shareholder].amount);
    }
}
```

### ● **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 set cooldown between buy transactions
- Fees can be set up to 50% for both buy and sell fees.
- Owner can exclude addresses from dividends.
- Owner can exclude addresses from fees.
- Owner can set addresses which are excluded from the timelock.
- The ownership of the contract isn't renounced.
- Owner is able to pause the contract
- Owner can set a max transaction amount.
- Owner can exclude addresses from max transaction amount constraint.
- Owner is able to blacklist contract addresses

# Contract Snapshot

```
contract FOREVERISE is IBEP20, Auth {
    using SafeMath for uint256;

    address WBNB = 0xbb4CdB9CBd36B01bD1cBaEaF2De08d9173bc095c;
    address DEAD = 0x00000000000000000000000000000000dEaD;
    address ZERO = 0x0000000000000000000000000000000000000000;

    string constant _name = "Foreverise";
    string constant _symbol = "FOREVER";
    uint8 constant _decimals = 9;

    uint256 _totalSupply = 100 * 10**9 * 10**_decimals;

    uint256 public _maxTxAmount = _totalSupply;
    uint256 public _maxWalletToken = _totalSupply;

    mapping (address => uint256) _balances;
    mapping (address => mapping (address => uint256)) _allowances;

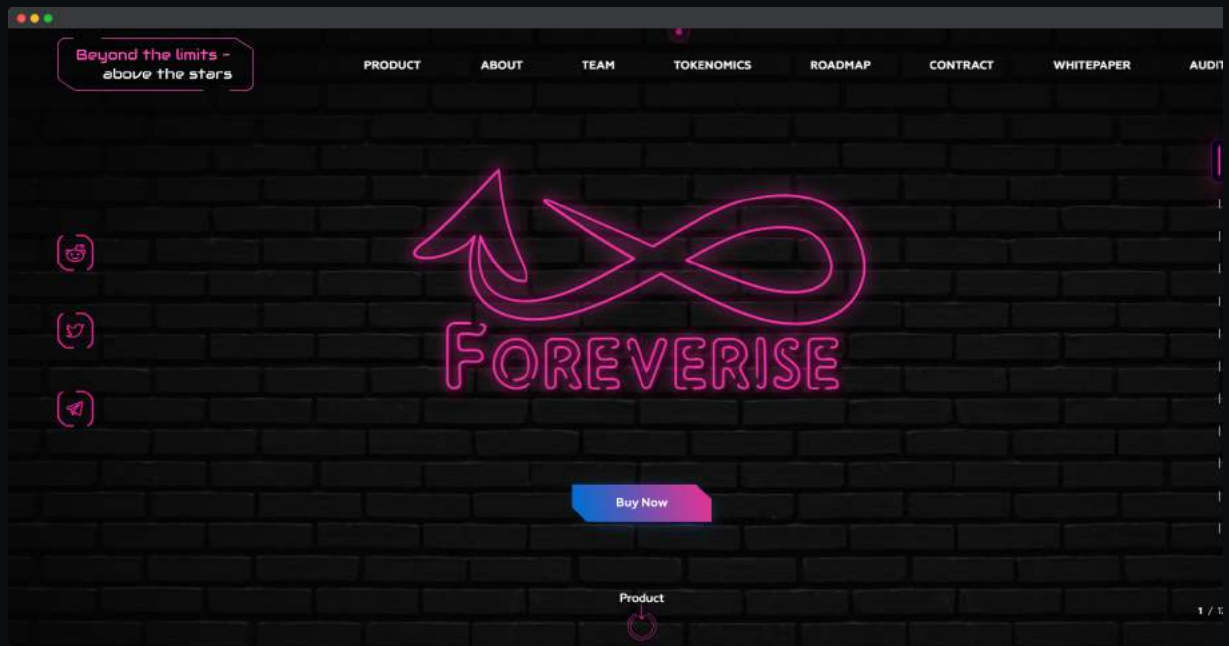
    bool public blacklistMode = true;
    mapping (address => bool) public isBlacklisted;

    mapping (address => bool) isFeeExempt;
    mapping (address => bool) isTxLimitExempt;
    mapping (address => bool) isTimelockExempt;
    mapping (address => bool) isDividendExempt;

    uint256 public liquidityFee = 2;
    uint256 public reflectionFee = 2;
    uint256 public marketingFee = 1;
    uint256 public teamFee = 0;
    uint256 public burnFee = 0;
    uint256 public totalFee = marketingFee + reflectionFee +
liquidityFee + teamFee + burnFee;
    uint256 public feeDenominator = 100;
```



# 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

Loading speed: 65%

# 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.