Vitalik Audit

Security Audit

Ezcoin Market

Saturday, 13 August 2022

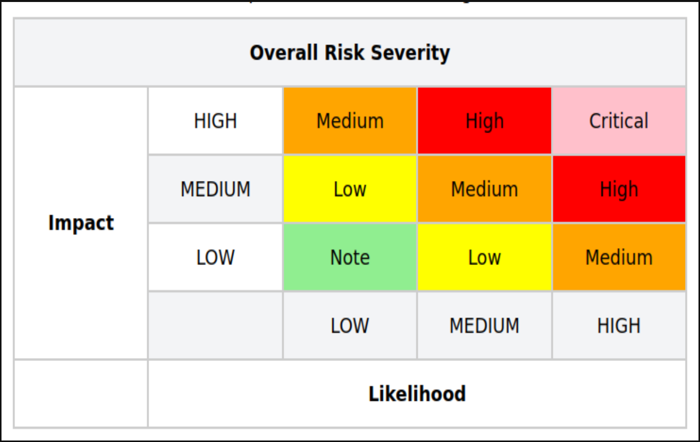
Severity Criteria

Vitalik assesses severity of disclosed vulnerabilities according to a methodology based on OWASP Standards.
Vulnerabilities are divided into 3 primary risk categories:

- Low
- Medium
- High

High-level considerations for vulnerabilities span the following key areas when conducting Assessments:

- Malicious Input Handling
- Escalation of privileges
- Arithmetic
- Gas use



Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Contract Address :
0x87e837803513e8c735a4eC1bcbf97Bc3050F41a5

Total Supply: 15,000,000 TKRN

Auditor: t.me/AnanCoder

Source Code SHA256 Hash:
024185b5f8f04567aa5ca960b1b9a8cd39020900ab08a0
74fa8c333d9e1a1fa6

Compiler version : v0.7.6+commit.7338295f

Audit Type : Manual + Automatic tools (launch testing)

Audit Date: 13/08/2022 17:51

Ezcoin Market – Overview

Concept & methodology

Ezcoinmarket Is Crypto Tracking Price For Crypto Asset and Free Signal Futures and Spot Trading

`Website : https://ezcoinmarket.com/`

Token Mechanism:

Design:
An RFI Token (Reflections), Forked from safemoon but not precisely like their contract because some functions got changed and there is also a couple of optimizations compared to safemoon.

Fees:
Currently, there is a 10% fee on buy and a 10% fee on sell. A portion of these fees will be collected inside the contract and, later on will get swapped to ETH and sent to 2 wallets:
-Promo wallet, which will be used for promoting and marketing purposes
-Shill wallet, we don't have information about the purpose of this wallet.
Another portion of the fees will be deducted from rTotal, which causes the token balance of holders to increase over time.

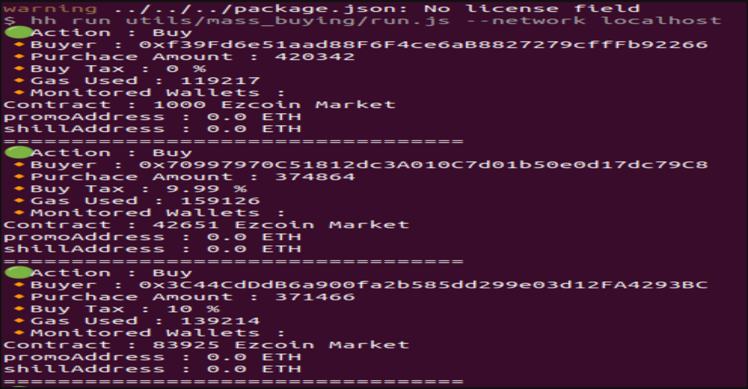
Owner Functions

excludeMultipleAccountsFromFees:
Used to exclude wallets from paying fees

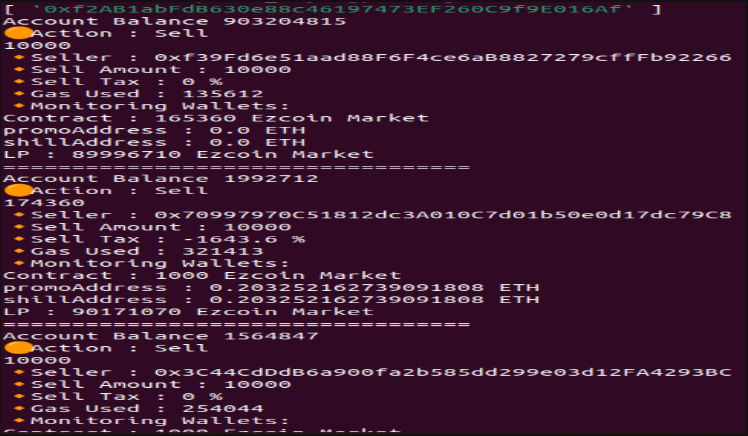
setFee:
Used to change buy or sell fees

Launch Test On Local Environment

Ezcoin Market launched on our local forked version of pancake swap, there was no problem with buying and selling and everything worked as expected



```
warning ../../../package.json: No license field
$ hh run utils/mass_buying/run.js --network localhost
🟢Action : Buy
 • Buyer : 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
 • Purchace Amount : 420342
 • Buy Tax : 0 %
 • Gas Used : 119217
 • Monitored Wallets :
Contract : 1000 Ezcoin Market
promoAddress : 0.0 ETH
shillAddress : 0.0 ETH
===================================
🟢Action : Buy
 • Buyer : 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
 • Purchace Amount : 374864
 • Buy Tax : 9.99 %
 • Gas Used : 159126
 • Monitored Wallets :
Contract : 42651 Ezcoin Market
promoAddress : 0.0 ETH
shillAddress : 0.0 ETH
===================================
🟢Action : Buy
 • Buyer : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
 • Purchace Amount : 371466
 • Buy Tax : 10 %
 • Gas Used : 139214
 • Monitored Wallets :
Contract : 83925 Ezcoin Market
promoAddress : 0.0 ETH
shillAddress : 0.0 ETH
===================================
```



```
[ '0xf2AB1abFdB630e88c46197473EF260C9f9E016Af' ]
Account Balance 903204815
🟠Action : Sell
10000
 • Seller : 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
 • Sell Amount : 10000
 • Sell Tax : 0 %
 • Gas Used : 135612
 • Monitoring Wallets:
Contract : 165360 Ezcoin Market
promoAddress : 0.0 ETH
shillAddress : 0.0 ETH
LP : 89996710 Ezcoin Market
===================================
Account Balance 1992712
🟠Action : Sell
174360
 • Seller : 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
 • Sell Amount : 10000
 • Sell Tax : -1643.6 %
 • Gas Used : 321413
 • Monitoring Wallets:
Contract : 1000 Ezcoin Market
promoAddress : 0.203252162739091808 ETH
shillAddress : 0.203252162739091808 ETH
LP : 90171070 Ezcoin Market
===================================
Account Balance 1564847
🟠Action : Sell
10000
 • Seller : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
 • Sell Amount : 10000
 • Sell Tax : 0 %
 • Gas Used : 254044
 • Monitoring Wallets:
```

Medium - Centralization Risk

Function : setFee

Line : 362

Description:

The owner can use this function to set taxes up to 99% percent, which can make investors unable to sell their funds. We believe that the Ezcoin team has proven their legitimacy by passing KYC and hosting their ICO on gempad, but having such functions can scare investors and question decentralization of the protocol.

Recommendation:

We recommend Ezcoin team to either remove this function (which is not possible considering the token design) or set a reasonable limit for taxes based on business logic

Medium – Bad Require Statement

Function : setFee

Line : 362

Description:

Require Statement is not matching the error message in this conditions:

```
require(redisFeeOnBuy < 100, "Redis cannot be
more than 10.");
require(redisFeeOnSell < 100, "Redis cannot be
more than 10.");
require(taxFeeOnBuy < 100, "Tax cannot be more
than 6.");
require(taxFeeOnSell < 100, "Tax cannot be
more than 6.");
```

Recommendation:

We recommend Ezcoin team to either remove this function (which is not possible considering the token design) or set a reasonable limit for taxes based on business logic

Low – Gas optimization & PK Risk

Function : ---
Line(s) : 161 - 162

Description:

Wallets _shillAddress & _promoAddress can not be changed after deployments. There is an important point that must be considered here.
These wallets can not be changed later, so if a wallet’s Private Key gets leaked, that would be an awful situation for the team as someone else has access to the marketing and development budget.

Recommendation

Create a function to be able to change this wallets later, if you don’t want to do so, consider changing this addresses to constant to save gas in deployment time.

Low – Potential Sandwich attack

Function : swapTokensForEth
Line(s) : 283

Description:

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can Manipulate the exchange rate by front running (before the transaction being attacked) a transaction to purchase one of the assets and make profits by back running (after the transaction being attacked) a transaction to sell the asset.

```
uniswapV2Router.swapExactTokensForETHSupportin
gFeeOnTransferTokens(
            tokenAmount,
            0,
            path,
            address(this),
            block.timestamp
);
```

Recommendation

give a reasonable output amount based on price

Low – Contract Balance

Function : ---
Line(s) : ---

Description:

Contract can accept both tokens and ether but there is not a withdraw function.

Recommendation

Create a function to withdraw contract’s ether and token balance

Low – Lack of return value handling

Function : swapTokensForEth
Line(s) : 283

Description:

Return value (true or false) is not being handled here

```
uniswapV2Router.swapExactTokensForETHSupportin
gFeeOnTransferTokens(
            tokenAmount,
            0,
            path,
            address(this),
            block.timestamp
);
```

Recommendation

We recommend using variables to receive the return value of the functions mentioned above and handle both success and failure cases if needed by the business logic

Optimization

Function : _transfer
Line(s) : 244

Description

```
if (from != owner() && to != owner())
```

this line causes whitelisted wallets to go through all the conditions in _transfer (same as a non-excluded wallet) this causes high gas usage

Recommendation:

remove that condition and add this condition at top of _transfer

```
if (_isExcludedFromFee[from]
_isExcludedFromFee[to]){
            _tokenTransfer(from,to,amount);
 }
```

this practice causes:

```
if ((_isExcludedFromFee[from] ||
_isExcludedFromFee[to]) ||
(from != uniswapV2Pair && to != uniswapV2Pair))
{
  _redisFee = 0;
  _taxFee = 0;
}
```

to be redunant and removeable which we will discuss in next page.

Optimization

Function : ---
Line(s) : 362, 373

Description

this functions should have external visibility since they are never used inside contract:

```
-setFee
-excludeMultipleAccountsFromFees
```

Recommendation:

Change visibility to external

Optimization

Function : _transfer

Line(s) : 244

Description

lines 263 - 273 are not gas optimized,the first if statement is checking whether from is address of pair(buying) whereas second if is checking the same scenario (for to), this causes more high gas usage

Recommendation:

change the statements to

```
if(from == uniswapV2Pair) {
   _redisFee = _redisFeeOnBuy;
   _taxFee = _taxFeeOnBuy;
}else if (to == uniswapV2Pair ) {
   _redisFee = _redisFeeOnSell;
   _taxFee = _taxFeeOnSell;
}
```

you can even remove the third if statement by simply doing

```
if(from == uniswapV2Pair) {
    _redisFee = _redisFeeOnBuy;
    _taxFee = _taxFeeOnBuy;
}else if (to == uniswapV2Pair ) {
    _redisFee = _redisFeeOnSell;
    _taxFee = _taxFeeOnSell;
}else{
    _redisFee = 0;
    _taxFee = 0;
}
```

this means if to and from are not pair address, then we are transfering between normal wallets

Small Optimizations

Line 135 :
remove Context from inheritance (redunant)

Line 139 :
_tOwned never used, delete it (redunant)

Lines 161-162:
make this addresses constant

Line 188 :
use address(0)

Line 247 :
remove this condition (redunant)

Disclaimer

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project orteam. This report is not, nor should be considered, an indication of the economics or value of any"product" or "asset" created by any team or project that contracts CertiK to perform a securityassessment.

This report does not provide any warranty or guarantee regarding the absolute bug-freenature of the technology analyzed, nor do they provide any indication of the technologies proprietors,business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with anyparticular project. This report in no way provides investment advice, nor should be leveraged as investmentadvice of any sort.

This report represents an extensive assessing process intending to help our customersincrease the quality of their code while reducing the high level of risk presented by cryptographic tokensand blockchain technology.

About Vitalik Audit

we are a small yet strong auditing company, we want to keep all prices ultra down but keeping quality up so that everyone is able to afford an audit.

Telegram: t.me/ContractCenter
Website: https://contractcenter.xyz
Owner: t.me/AnanCoder