Vitalik Audit

Security Audit

Token Runner

Thursday, 11 August 2022

Contract Address :
0xdB5f00592463A3F459f418eb831e4d2C4DB8E156

Total Supply: 15,000,000 TKRN

Auditor: t.me/AnanCoder

Source Code SHA256 Hash:
7553d03e68bf823b9df13019c878fb271746824561e823be0a63ba2c61d73bdb

Compiler version : v0.8.15+commit.e14f2714

Audit Type : Manual + Automatic tools (launch testing)

Audit Date: 11/08/2022 12:08

Token Runner – Overview

Concept & methodology

TokenRunner is created with centralized-decentralized attributes enhanced for entities, startups to create and launch their projects, putting investors first.

Website : https://tokenrunner.net

Token Mechanism:

Max 5% tax on buy and sell, collects this taxes and then swaps them to BNB (sent to marketing wallet), this marketing wallet will be used for promoting and marketing purpuse.

Total Score : 90 / 100

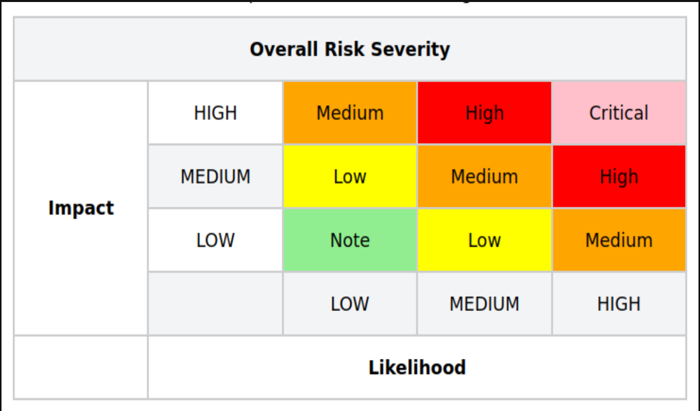
Severity Criteria

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			

Owner Functions

SetMarketingAddress:
used to change marketing address which collects taxes

SetSwapAndLiquifyEnabled:
Used to turn on/off contract collected tax swapping

SetSellMarketingFeePercent:
Used to change marketing fee on sell (can’t be more than 5%)

SetBuyMarketingFeePercent:
Used to change marketing fee on buy (can’t be more than 5%)

ExcludeFromFee:
Used to exclude addresses from paying fees

IncludeInFee:
Used to include addresses in fees

ChangeNumTokensSellToFee:
Changes swapping tershold

Note

Bad Contract Balance Handling

Line 803

Description:

if contractTokenBalance >= numTokensSellToFee then there is not need to assign numtokensSellToFee to contractTokenBalance. this may cause some of the tokens to be stucked in contract forever.

```
bool overMinTokenBalance = contractTokenBalance >=
numTokensSellToFee;
        if (
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            sender != uniswapV2Pair &&
            swapAndLiquifyEnabled
        ) {
            contractTokenBalance = numTokensSellToFee;
            //add liquidity
            swapAndLiquify(contractTokenBalance);
}
```

Recommendation:

Remove line 814

Note

Potential Sandwich Attacks

Line 877 - 880

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 manipulatethe exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one ofthe assets and make profits by backrunning (after the transaction being attacked) a transaction to sell theasset.

```
uniswapV2Router.swapExactTokensForETHSupport
ingFeeOnTransferTokens(
            tokenAmount,
            0, // accept any amount of ETH
            path,
            address(this),
            block.timestamp
);
```

Recommendation:

give a reasonable output amount based on price

Note

Lack of Event Emissions for Significant Transactions

Line 877 - 880

Description:

there are some functions that can change state variables, but they are not emitting an event, this can cause problems if you try to ingegrate your contract with an application later.

-setBuyMarketingFeePercent
-setSellMarketingFeePercent
-setMarketingAddress
-changeNumTokensSellToFee
-excludeFromFee
-includeInFee

Recommendation:

emit an event when a change happens

Note

Lack of Return Value Handling

Line 875

Description:

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

```
uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
            tokenAmount,
            0, // accept any amount of ETH
            path,
            address(this),
            block.timestamp
);
```

Recommendation:

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

Note

Contract Balance Handling

Description:

since contract can accept ether and tokens, we recommend you to add a function for withdrawing ether and tokens

Recommendation:

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

Gas Optimization

Functions Visibility

Description:

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

-includeInFee
-excludeFromFee
-setSwapAndLiquifyEnabled

Recommendation:

Change visibility to external

Gas Optimization

Redunant Lines

Description:

Line-636 - Redunant Code, this variable is provided by Ownable

Line-639 - Redunant Code, this assignment is done by Ownable

Line-640 - Redunant Code, this can be done with _mint

Line-642 - Redunant Code, instead use `_mint(msg.sender, amount)`

Lines-745 - required statement not matching error message

Lines-755 - required statement not matching error message

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**