Vitalik Security Audit

SecurePAY

Code Review + Launch Stimulation

Friday, 19 August 2022

Website: https://secure-pay.io

Twitter: https://twitter.com/SRPAY_SecurePay

Telegram: @SRPAY_Securepay

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			

Details

Name : SecurePay

Symbol: SRPAY

Total Supply: 100,000,000

Contract Address:

0x3C691f677408b100c4045e5b77ABA7F207389B14

Contract SHA256 Hash:

eab8b857d700c0eb6c41a493656201ee39fdcbc087cf2d72018ab450bb3636f8

Quick Overview & Findings

In terms of centralization there is no issues in the contract, consid -reing pinksale Safu criteria, owner of contract is not able to set taxes over 15% and is not able to **fully** disable or limit trading. However owner can set a max buy or sell amount but not less than 0.1% of total supply, this may limit whales from dumping price in **1 big trade.**
Owner is also able to set a max number of tokens for wallets, this means that wallets are not able to hold tokens more than that amount.(this limit can't be lower than 1% of total supply)

Overall, there is not and wont be any issues with buying, selling and transferring tokens **(SAFU)**

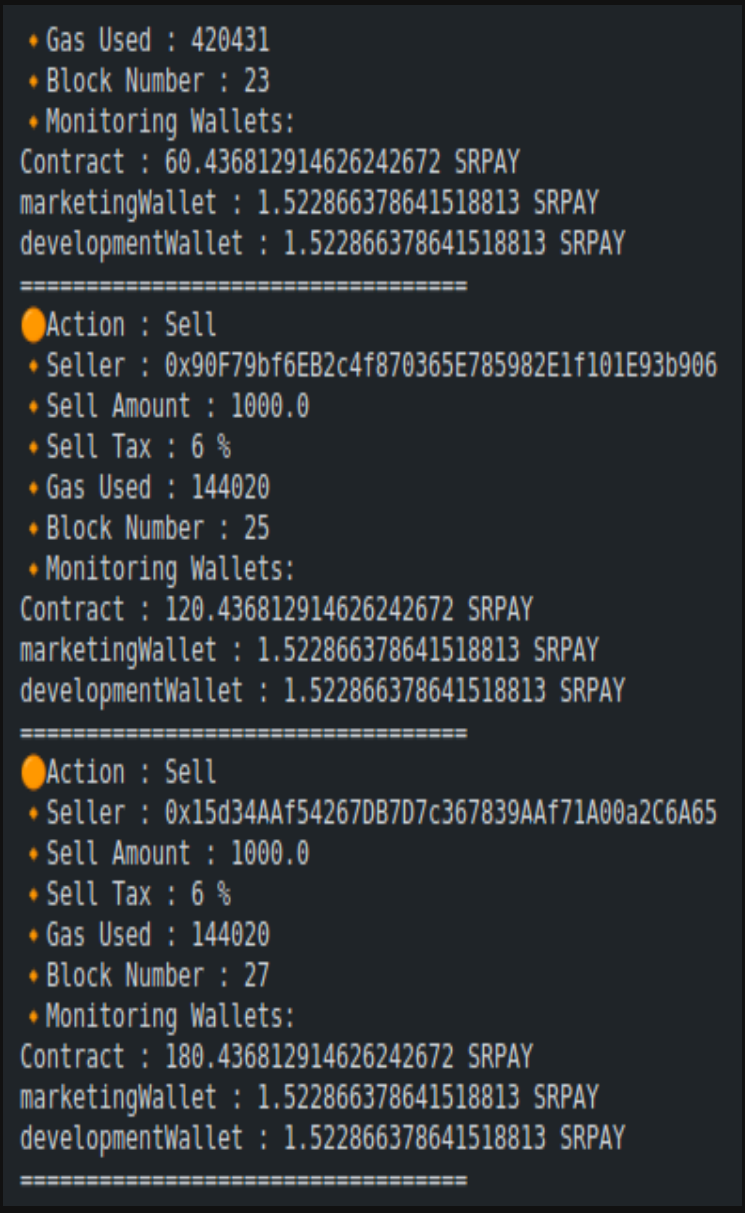
To make sure that everything goes right, we also launched $SRPAY on a testing blockchain and performed thousands of buys and sells, non of buys or sells due to contract errors, limits or high gas usage

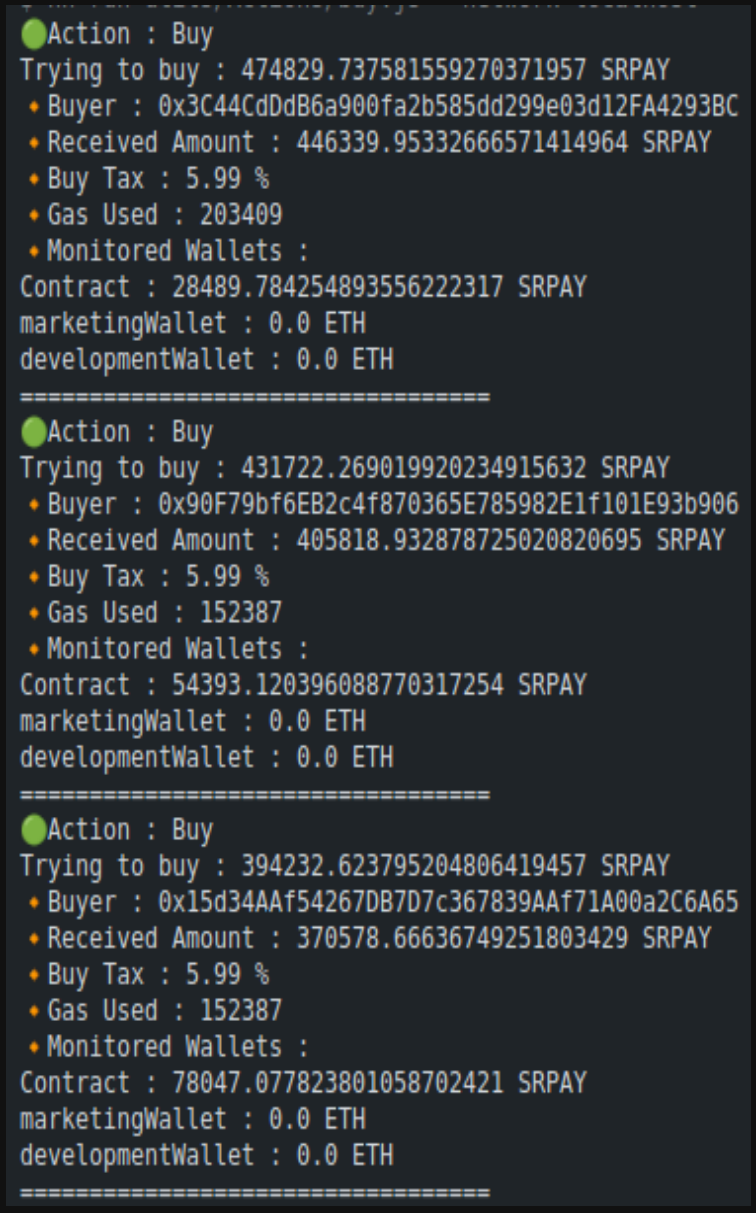
Findings:

- Low : 0
- Medium : 0
- High : 0
- Suggest : 5

Testing on local blockchain

First we added liquidity (SRPAY/WBNB) and then we used 10 testing accounts to buy and then sell $SRPAY.
taxes got collected inside contract and after reaching a limit, convert -ed into BNB and sent to marketing & development wallets





Selling & Converting

Buying

Findings – Suggestions

[O-01] – setAutomatedMarketMakerPair(address pair, bool value):

```
require(pair != uniswapV2Pair, "The PancakeSwap pair
cannot be removed from automatedMarketMakerPairs");
```

This condition must also check if value (input) is false or not. Otherwise we are not able to add new uniswapV2Pair to **automatedMarketMakerPairs**.

[O-02] - IUniswapV2Router02 public uniswapV2Router:

At line 463, define this variable as immutable to save gas

[O-03] - constructor (address newOwner, ...:

At line 497, you can define owner using msg.sender, passing owner address into constructor is redundant.

[O-04] – updateUniswapV2Router:

At line 570, check if newAddress is DEAD address or not.

[O-05] – handling potential sandwich attacks:

At line 765 and 818, when you are swapping, make sure to put a reasonable output amount, otherwise if trading amount is large, transactions may get front runned

Disclaimer

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team.

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 VITALIK to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature 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 any particular project. This report in no way provides investment advice, nor should be leveraged as Investment advice of any sort.

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

About Vitalik Audit

we are a small yet strong auditing team, we love to read and test smart contracts, if you think you are able to audit a smart contract or you want your project to be audited reach us through one of this ways.

Telegram: t.me/VitalikAudit
Website: https://vitalik-platform.xyz
Owner: t.me/AnanCoder