Vitalik Audit

Security Audit

GemDao

Monday, 15 Aug 2022

Severity Criteria

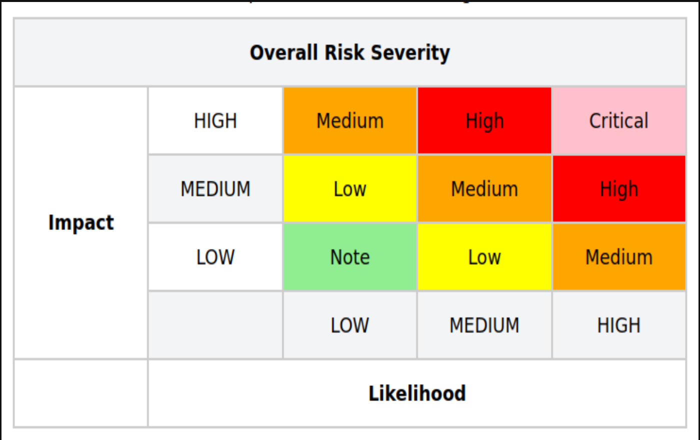
Vitalik assesses severity of disclosed vulnerabilities according to a methodology based on OWASP Standards.

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 :
0x3e990DE85Dbd92c9F616A1a4AbeAAE6243Be3
74b

Total Supply: 100,000,000 GemDao

Auditor: t.me/AnanCoder

Source Code SHA256 Hash:
049b7187be428caa606ae549abe9d5bdf22945ab74450
e94923bd4674673d69f

Compiler version : v0.8.7+commit.e28d00a7

Audit Type : Manual + Automatic tools (launch testing)

Audit Date: 14/08/2022 05:29

GemDAO – Overview

Concept & methodology

GemDAO is developed with team's desire to build an ecosystem to help the GemDAO's community easily evaluating, accessing, and investing in projects supported by GemDAO or community's choice

`Website : https://www.gemdao.io/`

Token Mechanism:

Design

GemDao token has 2 functionalities:
1- Limited trades in first 200 Blocks, this prevents high price volatility after launch which may lead to huge loses for some investors.
2- Accumulates taxes in contract and then sends them to an array of addresses defined by owner after reaching a threshold

Fees:

buy and sell fees can’t be more than 10% each (20% for buy + sell maximum)

Total Findings:

High : 0

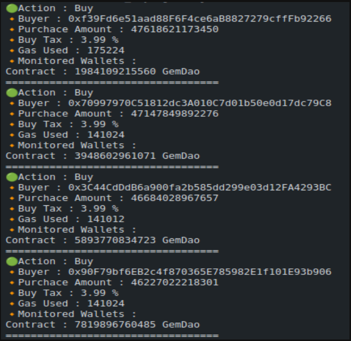
Medium : 0

Low : 3

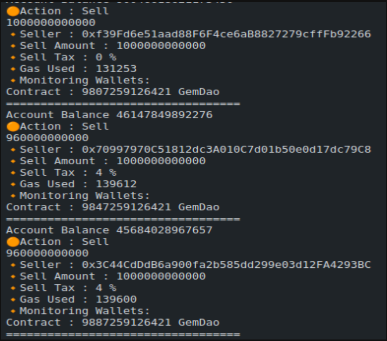
Info : 2

Main Features Tesed On Local Blockchain

Launching / Buying / Selling (Passed)
First we only added liquidity (WBNB) without making any changes to contract, buying and selling was fine with 4% tax on buy and 4% tax on sell.
As you can see in below pictures contract balance increased after each buy & sell.



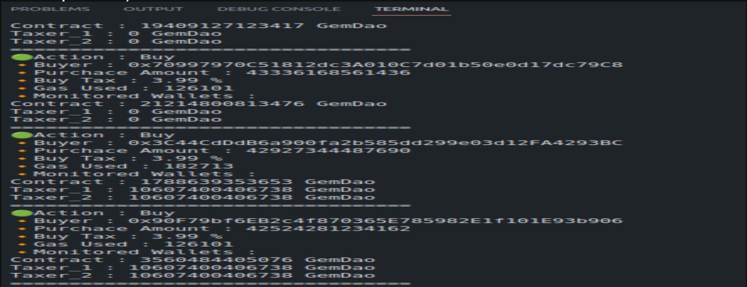
```
🟢Action : Buy
 • Buyer : 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
 • Purchace Amount : 47618621173450
 • Buy Tax : 3.99 %
 • Gas Used : 175224
 • Monitored Wallets :
Contract : 1984109215560 GemDao
=================================
🟢Action : Buy
 • Buyer : 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
 • Purchace Amount : 47147849892276
 • Buy Tax : 3.99 %
 • Gas Used : 141024
 • Monitored Wallets :
Contract : 3948602961071 GemDao
=================================
🟢Action : Buy
 • Buyer : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
 • Purchace Amount : 46684028967657
 • Buy Tax : 3.99 %
 • Gas Used : 141012
 • Monitored Wallets :
Contract : 5893770834723 GemDao
=================================
🟢Action : Buy
 • Buyer : 0x90F79bf6EB2c4f870365E785982E1f101E93b906
 • Purchace Amount : 46227022218301
 • Buy Tax : 3.99 %
 • Gas Used : 141024
 • Monitored Wallets :
Contract : 7819896760485 GemDao
=================================
```



```
🟠Action : Sell
1000000000000
 • Seller : 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
 • Sell Amount : 1000000000000
 • Sell Tax : 0 %
 • Gas Used : 131253
 • Monitoring Wallets:
Contract : 9807259126421 GemDao
==================================
Account Balance 46147849892276
🟠Action : Sell
960000000000
 • Seller : 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
 • Sell Amount : 1000000000000
 • Sell Tax : 4 %
 • Gas Used : 139612
 • Monitoring Wallets:
Contract : 9847259126421 GemDao
==================================
Account Balance 45684028967657
🟠Action : Sell
960000000000
 • Seller : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
 • Sell Amount : 1000000000000
 • Sell Tax : 4 %
 • Gas Used : 139600
 • Monitoring Wallets:
Contract : 9887259126421 GemDao
==================================
```

Anti-Bot (Passed)
for testing anti-bot we changed "pair" variable to address of WBNB/GemDao pair and we also changed anti-bot threshold to a reasonable amount based on our testing environment, every thing worked as expected meaning no one was able to sell more than threshold when token was in anti-bot block range, then we mined 200 blocks and after that selling was fine even above anti-bot threshold

Taxers (Passed)
we added couple of testing accounts as Taxers (a type of account in contract that takes the tokens which were collected inside contract), after reaching the tax threshold, tokens were successfully sent to those Taxers (as you can see in below picture)



```
Contract : 19409127123417 GemDao
Taxer_1 : 0 GemDao
Taxer_2 : 0 GemDao
=================================
🟢Action : Buy
 • Buyer : 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
 • Purchace Amount : 43336168561436
 • Buy Tax : 3.99 %
 • Gas Used : 126101
 • Monitored Wallets :
Contract : 21214800813476 GemDao
Taxer_1 : 0 GemDao
Taxer_2 : 0 GemDao
=================================
🟢Action : Buy
 • Buyer : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
 • Purchace Amount : 42927344487690
 • Buy Tax : 3.99 %
 • Gas Used : 182713
 • Monitored Wallets :
Contract : 1788639353653 GemDao
Taxer_1 : 10607400406738 GemDao
Taxer_2 : 10607400406738 GemDao
=================================
🟢Action : Buy
 • Buyer : 0x90F79bf6EB2c4f870365E785982E1f101E93b906
 • Purchace Amount : 42524281234162
 • Buy Tax : 3.99 %
 • Gas Used : 126101
 • Monitored Wallets :
Contract : 3560484405076 GemDao
Taxer_1 : 10607400406738 GemDao
Taxer_2 : 10607400406738 GemDao
=================================
```

Low – Adding Liquidity & Anti-bot

Function : antiBots

Line : 848

Description:

Setting pair (using setPair function) before adding liquidity can enable anti-bot, since anti-bot can only be used 1 time for 200 Blocks after starting, this issue may disable anti-bot in launch time if team doesn’t want to launch their token immediately after adding liquidity.

Recommendation:

Make sure to use setPair **after** adding liquidity

Low – Stuck Tokens Inside Contract

Function : transferTax

Line : 940

Description:

```
uint256 tax = amount.div(length);
```

The problem is that reminder of amount.div(length) may not be 0 (its not in most cases) so some of tokens will be stuck in contract, we saw this issue when we were testing the token (you can see this stuck amount in contract (testing page))

On the other hand, since contract has receive function, its possible for ether (BNB) to get stuck inside contract

Recommendation:

make 2 functions to be able to withdraw both GemDao as well as Ether Tokens

Low – Lack of Event Emission

Function : ---

Line(s) : ---

Description:

This functions are not emitting an event:
setTaxers, transferTax, takeTax, setPair, setAntiBotThreshold, setTaxThreshold, setTax, setExcludeTax, setExchanges

Recommendation:

Consider emitting a proper event for each one

Info – Setting Anti-Bot Threshold

Function : setAntiBotsThreshold

Line : 870

Description:

Not setting antiBotsThreshold to a reasonable number more than 0 before launch disables buying and selling for couple of blocks until owner change it successfully.

```
function antiBots(address to, uint256 amount) internal
virtual {
        if (startAntiBlock == 0) {
            if(to == pair) {
                startAntiBots();
            }
        } else if (block.number <
startAntiBlock.add(endAntiAfter)) {
require(amount < antiBotsThreshold, "ERC20: anti bots");
}
```

Recommendation:

Make sure to change this variable before launch.

Info – Taxers Length

Function : _transfer and transferTax

Line(s) : 911, 940

Description:

```
uint length = taxers.length;
uint256 amount = balanceOf(address(this));
uint256 tax = amount.div(length);
```

Not setting taxers before launching can revert all the transactions after reaching taxThreshold since we are dividing amount of collected tax by taxers length

Recommendation:

Make sure to initialize taxers before launch

Improvements & Optimizations

SafeMath Library:

Delete SafeMath from contract (unnecessary in compilers > 0.8.0)

Line 839:

No need to use Owner() again (redundant)

setPair function:

Add a dead address validation at setPair function

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