STAYSAFU

# AUDIT

*APRIL 3RD, 2023*

Fiero

# TABLE OF CONTENTS

# AUDIT SUMMARY

This report was written for Fiero in order to find flaws and vulnerabilities in the Fiero project's source code, as well as any contract dependencies that weren't part of an officially recognized library.

A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and Fiero Deployment techniques. The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors
- Assessing the codebase to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Through line-by-line manual review of the entire codebase by industry expert

# AUDIT OVERVIEW

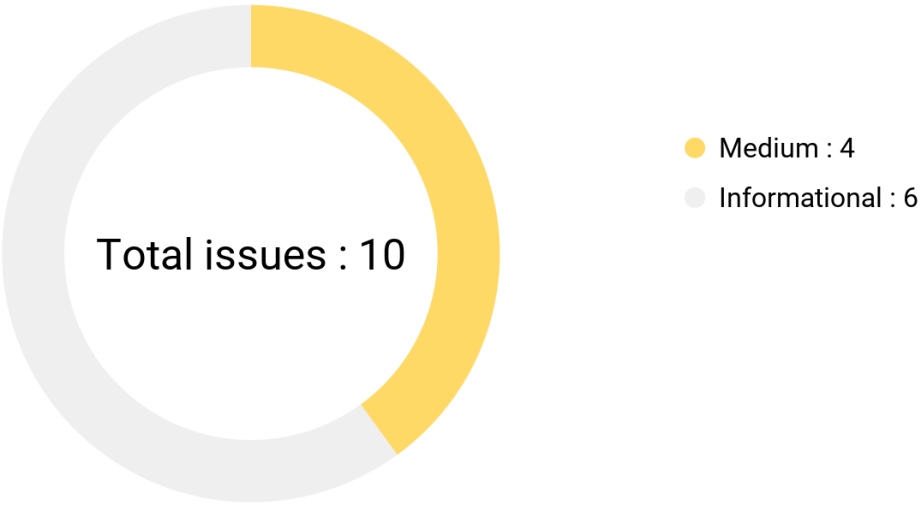
## PROJECT SUMMARY

| | |
|---|---|
| Project name | Fiero |
| Description | ... |
| Platform | ... |
| Language | Solidity |
| Codebase | - TokenSwapAgentImpl<br>- BEP20TokenImplementation<br>- CoinSwapAgentImpl |

## FINDINGS SUMMARY

| Vulnerability | Total |
|---|---|
| Critical | 0 |
| Major | 0 |
| Medium | 3 |
| Minor | 0 |
| Informational | 6 |

# AUDIT FINDINGS




| Code | Title | Severity |
|---|---|---|
| FEE-1 | setSwapFee | Informational |
| SWAP-1 | createSwapPair | Informational |
| SWAP-2 | fillCoin2TokenSwap | Medium |
| SWAP-3 | swapToken2Coin | Medium |
| LIST-1 | whiteList | Informational |
| LIST-2 | removeWhitelist | Informational |
| BLCK-1 | blacklist, removeBlackList | Informational |

| | | |
|---|---|---|
| FEE-2 | updateSwapFee | Informational |
| SWAP-4 | fillToken2CoinSwap | Medium |
| SWAP-5 | swapCoin2Token | Medium |

## File: TokenSwapAgentImpl

### FEE-1 | setSwapFee

#### Description

The owner can change the swap fee with this function.

#### Recommendation

This changes important state variables. An event is recommended after the process.

## SWAP-1 | createSwapPair

### Description

It's an owner-only contract that creates an upgradable BEP20 contract and initializes it.

### Recommendation

There are state variables written after the external calls. ReentrancyGuard is recommended.

## SWAP-2 | fillCoin2TokenSwap

### Description

Basically, this owner-only function mints the given token contract.

There is a check for coin transaction hash like this which reverts if the given bytes32 parameter is not flagged as used.

### Recommendation

You can't check if the hash is a valid transaction hash on-chain. The owner can use a different transaction hash for the same token and mint tokens again.

## File: BEP20TokenImplementation

### SWAP-3 | swapToken2Coin

#### Description

The function transfers the given tokens and burns them, then takes a fee from the account.

#### Recommendation

This isn't the only way to burn tokens. An account can burn the tokens in the original token contract itself and the accounts can do the same transaction without paying any fee.

## LIST-1 | whiteList

### Description

The function allows the owner to whitelist accounts.

### Recommendation

If an account (that's not whitelisted) gets flagged as a whitelist account, the account gets added to the nomineeList. There are no checks for duplicates so one account can get added to nomineeListmultiple times.

It's recommended to emit an event afterward.

## LIST-2 | removeWhitelist

### Description

The function allows the owner to remove an account from the whitelist, and nominee list.

### Recommendation

If an account has duplicates in nomineeList, the contract iterates over the list and removes the first encounter.

The internal function to find the index iterates over all nominee lists which might be pretty long. You can use Iterable Mappinglibraries to optimize time and gas complexity.

Just like whitelist, an event emit after the execution is recommended.

## BLCK-1 | blacklist, removeBlackList

### Description

This function allows the owner to blacklist accounts from transferring the tokens.

### Recommendation

An event emit after the execution is recommended.

## File: CoinSwapAgentImpl

### FEE-2 | updateSwapFee

#### Description

It's a basic function that changes the swap fee.

#### Recommendation

An event emit is recommended after the variable update.

## SWAP-4 | fillToken2CoinSwap

### Description

This function allows the owner to deposit some ethers for the exact amount of WFIERO. The function reverts if the given _tokenTxHash is already flagged as used in the filledTokenTx list.

### Recommendation

You can't verify if the transaction hash is real so the owner can use pseudo transaction hashes to use this function.

It's an owner-only function that only can be used by the owner, but the accounts can wrap ethers to WFIERO with WFIERO.deposit function. If the deposit is meant to be public you can skip this suggestion.

## SWAP-5 | swapCoin2Token

### Description

This function allows accounts (except contracts) to swap WFIERO for ethers. The swap fee is taken here.

### Recommendation

msg.value > swapFee checks are done multiple times. You don't have to check it in an if statement after you've checked the variable in the require statement.

This isn't the only way to unwrap WFIERO to ethers. The accounts can unwrap the WFIERO from its own contract without paying any gas fees.

# Global security warnings

## Allowances and Approves

Allowance checks are used in transferFrom and burnFrom functions.

```
function transferFrom(address sender, address recipient, uint256 amount) external override
returns (bool) {
    _transfer(sender, recipient, amount);
    _approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount,
"BEP20: transfer amount exceeds allowance"));
    return true;
}
```

There are no manual allowance checks with require, the contract uses SafeMath.sub to prevent the allowance from going below zero.

## Recommendation

The allowance should be checked before the _transfer process. The transaction will use a lot of gas before checking if the recipient got the allowance.

The contract does not allow the spender to exceed the allowance at this version, but a manual check with require is recommended in the contract.

# DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement.

This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement.

This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without StaySAFU's prior written consent.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 StaySAFU 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 tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk.

StaySAFU's position is that each company and individual are responsible for their own due diligence and continuous security. StaySAFU's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or fun.