**Final Notes: FrogeX**

**Brewlabs Services Hub**Summary: FrogeX and FXSwap are IERC20 ETH network token smart contracts for the FrogeX project. They are written in Solidity and are compiled using build 0.8.10. The FrogeX smart contract supports regenerative tokenomics.

Audit package: Standard security, with logic check

*Deployment Status:* Deployed to Ethereum Mainnet

##### *Files Audited:* FrogeX.sol

##### *Audit Result:* **Security PASSED, logic PASSED**

***Disclaimer:*** *This audit documentation is for discussion purposes only. The scope of this audit was to analyze and document FrogeX smart contract codebase for quality, security, and correctness. This audit guarantees that your code has been revised by an expert.*

**Overview of Audit**

##### No critical security issues identified.

##### 0 critical logic issues identified

##### 0 critical security issues identified.

##### 3 major security issues identified.

##### 0 major logic issues identified.

##### Few minor and informational issues identified across security and logic.

**Security issues**

Major level security issues detected:

##### The following issues violate the ethics of Decentralised Finance and present a centralisation risk to the community.

* 1. Contract allows owner to alter transaction fees.
  2. Contract allows owner to blacklist wallets and exclude wallets from rewards.
  3. Contract allows owner to remove liquidity. While there is a require function to prevent the liquidity from being removed prior to the *lockedUnlockDate* value, there is no logic in the contract to prevent the *lockedUnlockDate* from remaining as its initial declaration (which is the timestamp of contract deployment). Updating the value of the *lockedUnlockDate* is at the discretion of the owner.

These issues have been mitigated by the Development team, through the following actions:

- Contract owner is doxxed

- Contract events are emitted for the majority of state variable changes (eg. Line 484). These events which can be viewed by the community.

- Input validation on changes to state variables such as upper and lower boundary checks for fee values (eg. Line 552, 558).

Recommendation is for the Development team to continue Know Your Customer (KYC) services and communicate to community when relevant changes are committed to maintain transparency.

##### Minor level security issues detected:

##### Missing events for changes to state variables in various functions (Line 539, 543, 547, 615). This means only those authorised can alter state variables without the community being aware of changes. Recommendation is to add events for all changes to state variables or communicate to community when relevant changes are committed to improve transparency.

##### Informational issues detected:

##### Third party dependency on Uniswap Swap protocol (Line 199, 209, 211, 342, 497, 505). While it's fair to assume functional correctness of the protocol, developers in the space should be aware there is a possibility that third party entities can be compromised so it encouraged that teams monitor any future updates to the protocol and understand how they could impact the contract. Recommendation is for the Development team to monitor updates made to third party contracts to ensure changes do not impact the receiving contract or are mitigated/managed accordingly.

##### Contract does not support functionality to reinstate rewards or fees for wallets that have been previously excluded. While it’s understood this may be by design to support the business logic, recommendation is for the Development team to practice awareness when excluding wallets and educate any new contract owner that exclusions are final.

**Logic issues:**

Informational level logic issues detected:

* 1. FXSwap contract is prohibited from receiving any tokens from any sender other than WETH, which could present as limiting behaviour. This behaviour has been confirmed by the Development team who acknowledge that the contract will only need to receive WETH to support the business logic.
  2. Opportunity to condense codebase by using the getAmountsOut function presented in the UniSwapRouter interface since the implementation on line 186 appears to be a direct replica.

##### Naming conventions do not follow [Solidity guidelines](https://docs.soliditylang.org/en/v0.6.12/style-guide.html?highlight=order%20of%20layout#order-of-layout), Line 195, 296, 297, 298

##### Moderate use of comments. Comments help to improve readability and is encouraged. Refer to the [NatSpec Solidity](https://docs.soliditylang.org/en/develop/natspec-format.html) guidelines for further information, particularly for interfaces.

##### image2.png

**Conclusion:** The Brewlabs team thank you for the opportunity to review and audit your smart contact code. The data from this report will be formalised in the audit publication for your community. Keep in touch as we offer discounts for repeat business and on a range of other services!  
The Brewlabs Team.