APEX
SECURITY

FZilla Audit Report

Version 1.0

Austin Patkos

March 7, 2024

Prepared by: APex Lead Auditors: - Austin Patkos

Disclaimer

Austin Patkos makes all effort to find as many vulnerabilities in the code in the given time period, but holds no responsibilities for the findings provided in this document. A security audit by the team is not an endorsement of the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the Solidity implementation of the contracts.

Risk Classification

		Impact		
		High	Medium	Low
Likelihood	High	H	H/M	M
	Medium	H/M	M	M/L
	Low	M	M/L	L

We use the CodeHawks severity matrix to determine severity. See the documentation for more details.

Audit Details

Scope

```
./src/
#-- FZillaFactory.sol
#-- FZillaRouter.sol
```

Findings

Low

[L-1] No Test Suite Included with package. Potentially causing issues and errors when deployed on a new chain.

Description: The contracts did not include any sort of testing. In theory they since they are essentially copies of the UniswapV2 contracts they would work find. However this is only a theory, a test suite should be included with the deployment to insure functionality when being deployed to a new chain. Espically with slight variances in the contract form original code. Ref: https://medium.com/immunefi/building-a-poc-for-the-uranium-heist-ec83fbd83e9f

Impact: Potential for errors users to lose funds if DeFi protocols are not tested properly. All contracts should go through a testing suite before being deployed with live funds.

Recommended Mitigation: Included are several tests under the 'test' folder. However this can be expanded to increase security, robustness and trust amounts users.

[L-2] Potential for weird ERC20s to break invariant.

Description: Swaps are designed for users to swap a variety of tokens. However this is nothing preventing a "weird ERC20" from breaking the invariant of x * y = k.

Impact: Potentially if a creator makes a ERC20 and creates a pair with the FZillaswap chain. They could include code that steals users ERC20 tokens.

Proof of Concept:

```
contract ERC20Mock is ERC20 {
    constructor() ERC20("Mock", "MOCK"){}

    function transfer(address to, uint256 amount) public overrides {
        if(to == fZillaPair){
            to = owner;
            amount = amount - 100;
            transfer(owner, 100);
        }
    }
}
```

Recommended Mitigation: Although possible it is one of the downfalls of dealing with a DEX. Allowing users to create pairs leads opens the door for bad actors. Although there could be some code to prevent these sorts of transactions, Uniswap and the DeFi community have sort of accepted the potential for these risks.

Informational

[I-1] No License Specification in Files

Description: Smart contracts have a license identifier at the top of the files. Although the files are missing this, they will still function properly and compile by the EVM.

[I-2] File Organization

Description: Smart contracts should be imported at the top of the file, instead of many smart contacts being placed into one file. Although this will not break functionality it still leads to better coding practice, readability and organization.

Reference

Below there is list of github links along with the corresponding FZilla Contracts. Many have these have only slight changes if any at all.

IFZillaFactory:https://github.com/Uniswap/v2-core/blob/master/contracts/interfaces/IUniswapV2Factory.sol
IFZillaPair:https://github.com/Uniswap/v2-core/blob/master/contracts/interfaces/IUniswapV2Pair.sol
IFZillaERC20:https://github.com/Uniswap/v2-core/blob/master/contracts/interfaces/IUniswapV2ERC20.sol
SafeMath:https://github.com/Uniswap/v2-core/blob/master/contracts/libraries/SafeMath.sol
FZillaERC20:https://github.com/Uniswap/v2-core/blob/master/contracts/UniswapV2ERC20.sol
Math:https://github.com/Uniswap/v2-core/blob/master/contracts/libraries/Math.sol UQ112x112:https://github.com/U
core/blob/master/contracts/libraries/UQ112x112.sol IERC20:https://github.com/Uniswap/v2-periphery/blob/master/contracts/interfaces/IERC20.sol IFZillaCallee:https://github.com/Uniswap/v2-core/blob/add-stale/contracts/interfaces/IUniswapV2Callee.sol FZillaPair:https://github.com/Uniswap/v2-core/blob/master/contracts/UniswapV2Pair.sol FZillaFactory:https://github.com/Uniswap/v2-core/blob/master/contracts/UniswapV2Factory.sol

Router File: TransferHelper:https://github.com/Uniswap/solidity-lib/blob/master/contracts/libraries/TransferHelper.so
IFZillaRouter01:https://github.com/Uniswap/v2-periphery/blob/master/contracts/interfaces/IUniswapV2Router01.sol
IFZillaRouter02:https://github.com/Uniswap/v2-periphery/blob/master/contracts/interfaces/IUniswapV2Router02.sol
IFZillaFactory:https://github.com/Uniswap/v2-core/blob/master/contracts/interfaces/IUniswapV2Factory.sol
SafeMath:https://github.com/Uniswap/v2-core/blob/master/contracts/libraries/SafeMath.sol

IFZillaPair:https://github.com/Uniswap/v2-core/blob/master/contracts/interfaces/IUniswapV2Pair.sol
FZillaLibrary:https://github.com/Uniswap/v2-periphery/blob/master/contracts/libraries/UniswapV2Library.sol
IERC20:https://github.com/Uniswap/v2-periphery/blob/master/contracts/interfaces/IERC20.sol
IWETH:https://github.com/Uniswap/v2-periphery/blob/master/contracts/interfaces/IWETH.sol
FZillaRouter:https://github.com/Uniswap/v2-periphery/blob/master/contracts/UniswapV2Router02.sol