Security Audit Report

FRIENDLY MARKET

2022/07/18

PREPARED FOR:
Friendly Market, Friendly.Market

ARCADIA CONTACT INFO
Email: audits@arcadiamgroup.com
Telegram: https://t.me/thearcadiagroup

Table of Contents

Executive Summary

A Representative Party of **Friendly Market** (**"CLIENT"**) engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following **Friendly Market's** smart contracts on the **FriendlyMarket/friendly-amm-core**

The scope of this audit included the following files:

1. **friendly-amm-core/pair**
2. **friendly-amm-core/factory**
3. **friendly-amm-core/libs**
4. **friendly-amm-periphery/libs**
5. **friendly-amm-periphery/router**
6. **friendly-amm-periphery/wcspr**

Arcadia completed this security review using various methods primarily consisting of dynamic and static analysis. This process included a line-by-line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

There were **5** issues found, **1** of which were deemed to be 'critical', and **0** of which were rated as 'high'.

Severity Rating	Number of Original Occurrences	Number of Remaining Occurrences
CRITICAL	1	0
HIGH	0	0
MEDIUM	1	0
LOW	0	0
INFORMATIONAL	3	0

Findings

1. No need to store decimals into storage.

Issue: **FRIENDLY-MARKET**-1
Severity: **INFORMATIONAL**

Target: **core/pair/src/lib.rs**
Finding Type: **DYNAMIC**

In Uniswap V2, the **decimals** are constant values.
https://github.com/Uniswap/v2-core/blob/master/contracts/UniswapV2ERC20.sol#L11

And the factory contract uses 18 as a decimals for all pairs.
https://github.com/FriendlyMarket/friendly-amm-core/blob/main/factory/src/lib.rs#L1238

So the Pair contract no need to load decimals from storage, just return raw decimals (18 decimals)

Action Recommended:

Return constant value instead of loading decimals from storage.
https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L59-L62

2. UInt cannot be negative

Issue: **FRIENDLY-MARKET**-1
Severity: **INFORMATIONAL**

Target: **core/pair/src/lib.rs**
Finding Type: **DYNAMIC**

https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L418

The **liquidity** is a uint, so it cannot be negative.

The following lines have the same issue.

https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/checks.rs#L27

https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L569

https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L639

Action Recommended:

Change **less and equal** operator to **equal** operator

3. Wrong cumulative price calculation

Issue: **FRIENDLY-MARKET**-1
Severity: **CRITICAL**

Target: **core/pair/src/lib.rs**
Finding Type: **DYNAMIC**

To calculate cumulative price, the contract needs to multiply elapsed timestamp.
https://github.com/Uniswap/v2-core/blob/master/contracts/UniswapV2Pair.sol#L79-L80

However, the _update function in the pair does not multiply **timeElapsed**.
https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L1094-L1109

This makes it unable to calculate TWAP price, so it's difficult to make oracle on this AMM.

Action Recommended:

Multiple **timeElapsed** to the cumulative price values.

4. WCSPR symbol check is not enough for validation

Issue: **FRIENDLY-MARKET**-1
Severity: **MEDIUM**

Target: **core/pair/src/lib.rs**
Finding Type: **DYNAMIC**

https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L508-L517
In this line, it checks the token symbol, and if the symbol is WCSPR, then does a different function call for token transfer, however, this validation is not enough for WCSPR validation.

Now several tokens have the same symbols, so if there is any other token which has “WCSPR” as a symbol, then it is unable to burn liquidity.

Same issue in the following lines.
https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L598-L612
https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L718-L763
https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L869-L886

Action Recommended:

Store WCSPR token address and use it for validation.

5. No need to use the **pow** operator.

Issue: **FRIENDLY-MARKET**-1
Severity: **INFORMATIONAL**

Target: **core/pair/src/lib.rs**
Finding Type: **DYNAMIC**

https://github.com/FriendlyMarket/friendly-amm-core/blob/main/pair/src/lib.rs#L664-L666
1000 is a constant value, and 1000 ** 2 is a 1000000 and it’s also a constant value.

Action Recommended:

Multiple 1000000 instead of calculating 1000 ** 2

Conclusion

All issues Arcadia identified were remediated in a later commit.

Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.