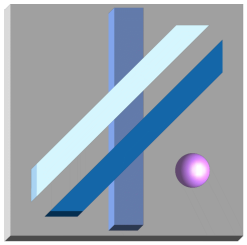

Fixed Point Solutions, LLC

market.xyz Fuse and Compound Changes Assessment

2021/10/01
Prepared by: Kurt Barry

1. Scope

The following diffs were reviewed by a single auditor over 10 hours of effort:

https://github.com/marketxyz/fuse-contracts/compare/b9d360d4110cd87bf21c4e596ea29be7b92c5569...9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb
https://github.com/marketxyz/compound-protocol/compare/8ea83bc180fd259ca80c09ef8f745c3221b15d39...1ffac350620ae79b874e24fbc32f387180570e2a
https://github.com/marketxyz/fuse-contracts/commit/e1dfbc8524029e889705ef7b95da135c81b89c00 (only partially reviewed due to time constraints)

2. Limitations

No assessment can guarantee the absolute safety or security of a software-based system. Further, a system can become unsafe or insecure over time as it and/or its environment evolves. This assessment aimed to discover as many issues and make as many suggestions for improvement as possible within the specified timeframe. Undiscovered issues, even serious ones, may remain. Issues may also exist in components and dependencies not included in the assessment scope.

3. Findings

Findings and recommendations are listed in this section, grouped into broad categories. It is up to the team behind the code to ultimately decide whether the items listed here qualify as issues

that need to be fixed, and whether any suggested changes are worth adopting. When a response from the team regarding a finding is available, it is provided.

Findings are given a severity rating based on their likelihood of causing harm in practice and the potential magnitude of their negative impact. Severity is only a rough guideline as to the risk an issue presents, and all issues should be carefully evaluated.

Severity Level Determination		Impact		
		High	Medium	Low
Likelihood	High	**Critical**	**High**	**Medium**
	Medium	**High**	**Medium**	**Low**
	Low	**Medium**	**Low**	**Low**

Issues that do not present any quantifiable risk (as is common for issues in the Code Quality category) are given a severity of **Informational**.

3.1 Security and Correctness

Findings that could lead to harmful outcomes or violate the intentions of the system.

None.

3.2 Usability and Incentives

Findings that could lead to suboptimal user experience, hinder integrations, or lead to undesirable behavioral outcomes.

U.1 Potentially Unexpected Behavior for Zero Address Arguments in `exchangeAllTokens` Could Lead to Losses for Integrators

Severity: **Low**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L86

Description: In the pre-modification code, a zero address was used as a stand-in for the native underlying blockchain asset (ETH, although on Polygon this would be MATIC). The modified code seems to have largely removed this feature. Integrators expecting similar behavior to that seen on mainnet might, however, accidentally pass the `exchangeTo` argument to `safeLiquidate` or `safeLiquidateTokensWithFlashLoan` as the zero address. Since

`exchangeAllTokens` will silently return without making any swap to the native asset, it is possible for callers to receive nothing if `minOutputAmount` was set to zero. It likely makes more sense to revert instead of return in the case that at least one of `from` or `to` is zero. This check should also be moved before the `from == to` check since otherwise if both `from` and `to` are `address(0)` (possible using a custom redemption strategy), the silent return behavior still results.

Response: Fixed in commit [fa4067733f9be37f76b0de3d5a4a1858379a6ab2](fa4067733f9be37f76b0de3d5a4a1858379a6ab2).

U.2 User Experience Could Be Worsened By Returning Less Information from `getTokenNameAndSymbol`

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/e1dfbc8524029e889705ef7b95da135c81b89c00/contracts/FusePoolLensV2.sol#L45

Description: In the original FusePoolLens contract, this function computed human-friendly symbols for LP tokens based on their constituent tokens. In FusePoolLensV2, this is removed. This could decrease UI performance as up to four additional RPC calls may be necessary, depending on how efficiently the UI is implemented, particularly if many LP tokens symbols are displayed on the same page.

Response: Acknowledged; UI uses caching to mitigate.

3.3 Gas Optimizations

Findings that could reduce the gas costs of interacting with the protocol, potentially on an amortized or averaged basis.

G.1 Use of `.div()` Is Less Efficient Than `/`, Which Behaves Identically

Severity: **Low**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L230

Description: OpenZeppelin's `div(a, b)` SafeMath function behaves no differently than Solidity's `/` operator, but is more expensive gas-wise.

Response: Acknowledged.

G.2 Extra Bytecode and Computational Load from Checked Arithmetic in Solidity 0.8.X

Severity: **Informational**

Code Location:
[1] all `++` operators in
https://github.com/marketxyz/fuse-contracts/blob/e1dfbc8524029e889705ef7b95da135c81b89c00/contracts/FusePoolUsersLens.sol
[2] all `++` operators in
https://github.com/marketxyz/fuse-contracts/blob/e1dfbc8524029e889705ef7b95da135c81b89c00/contracts/FusePoolLensV2.sol

Description: In code upgraded to Solidity 0.8.X, previously unchecked arithmetical operations become checked for overflow (or underflow). Since the functions in the upgraded contracts are not meant to be called on-chain, the impact is extremely minimal--slightly higher deployment costs than strictly necessary, and slightly more computational load placed on nodes being queried.

Response: Acknowledged.

3.4 Code Quality

CQ.1 `FuseSafeLiquidataor.repayWethFlashLoan()` Is Unused

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L265

Description: This function is private and has no callsites. Unless logic using it was accidentally deleted, it can be removed.

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.2 Could Use Return Variable in `transferSeizedFunds`

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L148

Description: Instead of declaring `seizedOutputAmount` as a stack variable and explicitly returning it, instead it could be declared as part of the function declaration (ie. `... internal returns (uint256 seizedOutputAmount)`) which would allow omitting both the stack declaration and explicit return statement.

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.3 "Uniswap" Instead of "Sushiswap"

Severity: **Informational**

Code Location:
[1]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L180
[2]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L187

Description: As noted, SushiSwap is used instead of Uniswap on Polygon; it would be good to keep this consistent in the comments.

Response: Acknowledged.

CQ.4 References to ETH Should Be To MATIC Instead

Severity: **Informational**

Code Location:
[1]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L153
[2]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L157
[3]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L213

Description: In these contexts MATIC is being manipulated, not ETH.

Response: Acknowledged.

CQ.5 Inconsistent Use of `isContract` Method

Severity: **Informational**

Code Location:
[1]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L217

[2]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L395

Description: In [1] the function is invoked via Solidity's type extension syntax, while in [2] it is invoked from its library. Of course the two are equivalent in functionality, but self-consistency of code is beneficial for readabilty and maintainability

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.6 Comment Refers to Nonexistent Function

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L61

Description: `postFlashLoanWeth` is no longer the name of any function in this contract.

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.7 Inconsistent Indentation

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L125

Description: This line of code and the ten following lines are indented further than other code within the same block.

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.8 Spelling/Grammar/Typos

Severity: **Informational**

Code Location:
[1]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L61
[2]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L330

Description:

[1] extra “/” character before “ after”
[2] “calling redeeming” → “redeeming”

Response: [1] fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.9 Unnecessary Separate Deserialization of Parameter in `uniswapV2CAll`

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L224

Description: This parameter could be decoded with all the others as part of the next statement, simplifying the code. Depending on compiler behavior, this may potentially save gas as well.

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.10 Inaccurate Comment Reference to Miner Tipping

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L202

Description: This comment mentions that the code will “send ETH to coinbase if necessary” but this logic has been removed.

Response: Fixed in commit fa4067733f9be37f76b0de3d5a4a1858379a6ab2.

CQ.11 References to “zero address if/for ETH” Are No Longer Accurate

Severity: **Informational**

Code Location:
[1]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L79
[2]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L80
[3]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L104
[4]https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L177

Description: This functionality has been removed.

Response: Fixed in commit [fa4067733f9be37f76b0de3d5a4a1858379a6ab2](fa4067733f9be37f76b0de3d5a4a1858379a6ab2).

CQ.12 MATIC Transfer Code Path in `exchangeAllTokens` Can Be Removed

Severity: **Informational**

Code Location:
https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L151

Description: Use of the native blockchain asset has been almost entirely removed except for in this function; unless there's a need to keep this feature, it could be removed entirely.

Response: Fixed in commit [5754c6f1a4ce7ef115022a966b28b85ad871c269](5754c6f1a4ce7ef115022a966b28b85ad871c269).

4. Notes

This section contains general considerations for interacting with or maintaining the system and various conclusions reached or discoveries made during the course of the assessment. Whereas findings generally represent things for the team to consider changing, notes are more informational and may be helpful to those who intend to interact with the system.

4.1 FuseSafeLiquidator.exchangeAllTokens Is Hardcoded to Route Swaps Via WETH

https://github.com/marketxyz/fuse-contracts/blob/9a8b159d5ea91673f9c70678f6dd7d9a62f3e8eb/contracts/FuseSafeLiquidator.sol#L94

If one or both of the tokens has low WETH liquidity on Polygon, high slippage could result. Integrators should be made aware of this consideration.