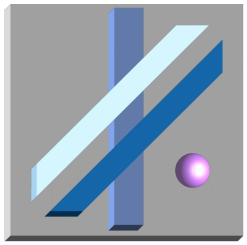

Fixed Point Solutions, LLC

Rari Vaults Assessment - Core

2021/11/21
Prepared by: Kurt Barry

1. Scope

The following files and directories were audited based on the linked commits:
https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol
https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/VaultFactory.sol

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.

SC.1 Critical Parameters Could Be Checked Explicitly in `initialize()`

Severity: Low

Code Location:
https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L858

Description: The comments above `initialize()` specify that critical parameters must be set before it is called. Explicit checks could be added to the function's logic for these parameters as an added safety measure (e.g. `require(harvestDelay > 0)`).

Response: Acknowledged.

SC.2 Replacing and Swapping Strategies in the Withdrawal Queue is Permissionless

Severity: Medium

Code Location:
[1]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L752
[2]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L780

Description: The `replaceWithdrawalQueueIndex` and `swapWithdrawalQueueIndexes` functions can be called by anyone, allowing arbitrary replacements and reorderings of

strategies. A malicious entity could use this to achieve various ends, for example selectively removing all funds from a particular strategy before another in violation of the intent of governance. These manipulations could take place within a single transaction, precluding any possibility of fixing the sabotage. The worst-case impact is difficult to assess because it might depend on the nature of the strategies and their interactions with other protocols. Since other operations that modify the withdrawal queue require authorization of the caller, it makes sense to apply the same restriction to these functions.

Response: Fixed in commit a44f067f598aecd43a65cbcfc8680d928fbd6bde.

SC.3 [Client-Reported] Logical Errors in `seizeStrategy`

Severity: Medium

Code Location:
[1]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L813
[2]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L816

Description: These errors, reported by the client before the audit progressed to this part of the contract, would have resulted in accounting issues.

Response: Fixed in commits 64a47ac91e4eb9391a25a2da977f4f216317b2e2 and 031c776193439f19ed591037d97202750fc6e3e0.

SC.4 [Partially Client-Reported] `harvest()` Incorrectly Assumes maxLockedProfit Is Zero

Severity: Medium

Code Location:
[1]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L452
[2]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L466

Description: The `harvest` function assumes that `maxLockedProfit` can be treated as zero, but this is only true for the first call to `harvest` with a harvest window. This assumption is used when computing fees [1] and when updating `maxLockedProfit` [2]. These calculations should be corrected to properly account for `maxLockedProfit`. In particular, the second allows an attacker to set locked profit to zero and use a flash loan to extract most of it for themselves during a harvest window.

Response: Fixed in commits 53ba195d04e51e0b0f39385f1a3ae82340d5c450 and ba3e59f24222042b00574ed520721aa6fe8e07f2.

SC.5 Calls Can Be Made to Untrusted Strategies

Severity: Medium

Code Location:
[1]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L692

Description: Discussion of the `Vault` contract's authorization model with the client revealed that the entities authorized to modify the withdrawal queue might differ from those that control whether particular strategies are trusted. Thus, a malicious withdrawal queue operator can insert untrusted strategies into the queue, which the `Vault` will make an external call to in the internal `pullFromWithdrawalQueue` function. A simple form of abuse would be to insert a strategy that always reverts, blocking all withdrawals that exceed the available amount of float until a more privileged actor steps in and rectifies the situation. Other mischief is possible, for example:

1. Arrange the withdrawal queue so that the first strategy to be pulled from is trusted, but the second is malicious;
2. the malicious strategy invests in the Vault;
3. when a user makes a withdrawal large enough to fully drain the first strategy and a call is made to the malicious one, it calls `withdraw` or `redeem`;
4. the exchange rate calculated will be too favorable, because `totalStrategyHoldings` has not yet been decreased, and because the underlying tokens have not yet been transferred to the user who is withdrawing (so the measured float is also erroneously large), allowing the malicious strategy to effectively steal from other depositors.

It is recommended that calls only be made to trusted strategies. Separately, it may be worth preventing reentrancy at least into the `deposit`, `withdraw`, and `redeem` functions as a further safety measure.

Response: Fixed in commit ba8c5a37fddf1b717361d2962b33fef1b7a83156.

3.2 Usability and Incentives

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

None.

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 Operations That Could Be Unchecked

Severity: **Low**

Code Location:
[1]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L325
[2]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L439

Description:
[1] This subtraction is within a conditional block that guarantees it will not underflow.
[2] This subtraction is in the branch of the ternary operator expression that guarantees it will not overflow.

Response: Acknowledged.

G.2 Redundant Variable in `pullFromWithdrawalQueue`

Severity: **Low**

Code Location:
https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L662

Description:
The `startingIndex` variable is only used to set the `currentIndex` variable and could be eliminated.

Response: Fixed in commit ba8c5a37fddf1b717361d2962b33fef1b7a83156.

3.4 Code Quality

CQ.1 Typos

Severity: **Informational**

Code Location:
[1]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L123

[2]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L358
[3]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L437
[4]https://github.com/Rari-Capital/vaults/blob/de132f05f1354715d95c8357ce3de5b102510c95/src/Vault.sol#L561

Description:
[1] Repeated word ("delay").
[2] "diving" → "dividing" (or conjugate as "divving" to avoid confusion with dive/diving)
[3] Repeated word ("it").
[4] Missing word ("it").

Response: Fixed in commit 5bfaca10002ef913df561d417470033260fd11cb.

CQ.2 Inconsistent Naming

Severity: **Informational**

Code Location: throughout Vault.sol

Description: Sometimes the Vault's share token is referred to as "rvToken", other times as "fvToken".

Response: Fixed in commit d0610cd0fe6dfb08baf70484637768f87eba7bc9.

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 Potential Share Price Manipulation

The price of vault shares can be manipulated by sending a quantity of the underlying token directly to a vault. In a situation where the total value of deposits in a vault is small, it might be economically feasible to change the share price significantly within a single transaction. This poses no risk to depositors, but other protocols that rely on the price of vault share tokens, especially lending protocols, should be aware of this possibility and take measures to ensure it poses no risk to their own users.