	Hats Finance Audit Competition Feb 14 - 27, 2023. Fuji Finance V2. See announcement					
#	Committee decision	Time (Arizona MST)	Submission title	Submission Description	GitHub issue Link	Grouped Categories
1	Low	2/14/2023 5:16	oracle: attacker can borrow more than he allowed to	Chainlink oracle price could return negative value.	https://github.com/Fujicracy/fuji-v2/issues/294	Oracle data verification
3	Medium	2/14/2023 6:52	Attacker can withdraw asset without burning any shares	After a lot of checks, function withdraw() calls to the previewWithdraw(assets) to get the shares coressponding to the amount of assets that the user want to withdraw. The vulnerability of this calculation is previewWithdraw() use the round-down instead of round-up.	https://github.com/Fujicracy/fuji-v2/issues/293	vault minAmount in withdraw
4	Low	2/14/2023 7:15	Unsafe usage of ERC20 .transfer()	The ERC20.transfer() and ERC20.transferFrom() functions return a boolean value indicating success. This parameter needs to be checked for success. Some tokens do not revert if the transfer failed but return false instead.	https://github.com/Fujicracy/fuji-v2/issues/318	safe erc20 transfer
5	Low	2/14/2023 7:58	Wrong implementation of BaseVault.previewMint	The function calculates the debtShares corresponding to the amount which the user want to borrow by a round-down calculation. This could lead to the vulnerability that when user borrow many times and try to payback in 1 turn, their transaction can be revert because of the folLowing condition:	https://github.com/Fujicracy/fuji-v2/issues/296	ERC4626-inflation attack
6	Low	2/14/2023 8:19	Oracle data can be outdated	A strong reliance on the price feeds has to be also monitored as recommended on the Risk Mitigation section. There are several reasons why a data feed may fail such as unforeseen market events, volatile market conditions, degraded performance of infrastructure, chains, or networks, upstream data providers outage, malicious activities from third parties among others.	https://github.com/Fujicracy/fuji-v2/issues/297	Oracle data verification
7	Low	2/14/2023 13:05	Wrong implmentation of function BorrowingVault. convertDebtToShares	The function calculates the debtShares corresponding to the amount which the user want to borrow by a round-down calculation. This could lead to the vulnerability that when user borrow many times and try to payback in 1 turn, their transaction can be revert because of the folLowing condition:	https://github.com/Fujicracy/fuji-v2/issues/305	vault round-up round-down
8	Medium	2/14/2023 14:47	TimeLock can prevent ERC-4626 compliance due to setting a Lowered deposit cap	BaseVault.sol has a function which alLows the timelock to input a deposit cap, setDepositCap(uint256). The function disalLows setting the cap as zero or beLow the minimum vault amount, however, it does not prevent the timelock from depositing an amount Lower than currently stored in the vault.	https://github.com/Fujicracy/fuji-v2/issues/307	changing depositCap checks
9	Low	2/15/2023 1:53	Vaults can be unable to pause or unpause due to DoS gas limit	_changePauseState() method loops over all the vaults to change their pause state. If there are a large number of vaults, this method may exceed the gas limit and fail to execute. as similar to this setSafetyRating() is internally used _checkValidVault() function and it also loop over the _vaults. as mentioned above setSafeRating() is also will not able to execute. if setSafetyRating() will not work then deployVault() will not work because it depends on it.	https://github.com/Fujicracy/fuji-v2/issues/309	pausing multiple vaults
10	High	2/15/2023 7:51	Incorrect gainedShared calculation when call function	Function FujiOracle._getUSDPrice() requests the price from chainlink oracle with a given asset. The chainlink then returns the price of 1 "real" token asset (not 1 wei) in usd which means with 10^asset_decimal asset how much usd you can get. From that definition, the function FujiOracle.getPriceOf(address currencyAsset, address commodityAsset, decimals) returns the value	https://github.com/Fujicracy/fuji-v2/issues/311	liquidate function decimal error
11	Low	2/16/2023 5:23	attacker can steal users assets	Attacker can stole user assets by deploying vault using malicious token because fuji is not checking valid assets or debtAsset. deployVault() has permissionlessDeployments, so when it is true anyone can deploy vaults.	https://github.com/Fujicracy/fuji-v2/issues/314	unusual erc-20 token interaction
13	Medium	2/16/2023 8:54	Inflation attack with BorrowingVault.sol when the provider is Aave2	The ERC426 vault is vulnerable to 1 type of attack called inflation attack. You can read about it here. It can apply to our BorrowingVault.sol contract when the provider is AaveV2.sol.	https://github.com/Fujicracy/fuji-v2/issues/315	ERC4626-inflation attack
14	Low	2/16/2023 9:34	Lack of supporting for Fee-on-Transfer token	There are ERC20 tokens that may make certain customizations to their ERC20 contracts. One type of these tokens is deflationary tokens that charge a certain fee for every transfer() or transferFrom(). Assume that the BaseVault.asset() is a deflationary one. When a user call function BaseVault.deposit(x, addr), the actual amount of asset tokens that contract received will be smaller than x. This will make the call _executeProviderAction(assets, "deposit", activeProvider) revert since the provider requires the contract transfer exactly x tokens.	https://github.com/Fujicracy/fuji-v2/issues/316	unusual erc-20 token interaction
15	Medium	2/16/2023 17:26	Timelock can increase max LTV to >100%, enabling undercollateralized loans	The borrowing vault has functions which modify it's risk parameters, one of it's functions being setMaxLtv() which has a Lower bound of 1%, the function however does not have any upper bound, meaning the timelock has the ability to increase the loan-to-value amount to above 100%. In the scenario where this happens, loans could be undercollateralized, making it profitable to borrow as many times as possible and leave the protocol with a terrible bad debt. Any value above 100% will cause this vulnerability	https://github.com/Fujicracy/fuji-v2/issues/320	setting value restriction maxLTV
17	High	2/17/2023 5:00	Attacker can deposit on behalf of users	The BaseRouter contract alLows bundling multiple actions in one call to execute them in one transaction. For deposit and payback actions, it will pull tokens from sender to this contract.	https://github.com/Fujicracy/fuji-v2/issues/323	excess approval to router - validate vault address input

	Hats Finance Audit Competition Feb 14 - 27, 2023. Fuji Finance V2. See announcement					
#	Committee decision	Time (Arizona MST)	Submission title	Submission Description	GitHub issue Link	Grouped Categories
18	Low	2/17/2023 5:57	Receive()` function is dangerous	It is possible for any user to send ether to `YieldVault.sol` contract by `receive` function, this ether sent to the contract will be locked up forever because there isn't any function to return miss sending ether.	https://github.com/Fujicracy/fuji-v2/issues/324	hanlde random send native token in vault
19	Low	2/17/2023 10:10	ERC20 transfer zero amount can be reverted	Certain ERC-20 tokens do not support zero-value token transfers and revert. As ERC20 can be an arbitrary token, in the case when such token doesn't alLow for zero amount transfers. This may break systems	https://github.com/Fujicracy/fuji-v2/issues/325	erc20 zero token transfer check
20	Low	2/17/2023 10:29	Malicious user can Blacklists Token	Some tokens (e.g. USDC, USDT) have a contract level admin controlled address blacklist. If an address is blocked, then transfers to and from that address are forbidden.	https://github.com/Fujicracy/fuji-v2/issues/326	blacklisted addresses in erc20 token type
21	1st Place Gas	2/17/2023 21:57	Gas Optimization	Private and will make public once competition ends	https://github.com/rotcivegaf/fuji-v2/compare/origin-v0.0.1...gas-saving	Gas
21	Low	2/17/2023 21:57	Unchecked transfer on `FlasherEuler.sol`	The return value of an external transfer call is not checked	https://github.com/Fujicracy/fuji-v2/issues/303	safe erc20 transfer
21	Low	2/17/2023 21:57	Unchecked transfer on `BaseRouter.sol`	The return value of an external transfer call is not checked	https://github.com/Fujicracy/fuji-v2/issues/302	safe erc20 transfer
22	Low	2/18/2023 18:58	Rewards token of vaults from providers are wasted	Some providers distribute rewards for users supplying or borrowing from them. But Fuji vaults doesn't have any function to recover these reward tokens. So the rewards claimed to the vaults will be wasted.	https://github.com/Fujicracy/fuji-v2/issues/329	missing harvesting functions
23	Medium	2/18/2023 20:16	Missing transfer fees from contract flasherBalancer to flashloan source of Balancer, then flasher can't flashloan to Balancer's vaults which have flashloan fee > 0		https://github.com/Fujicracy/fuji-v2/issues/330	missing fee payment in balancer flasher
24	High	2/21/2023 2:21	FujiOracle will return the wrong price for asset if underlying aggregator hits minAnswer	Chainlink aggregators have a built in circuit breaker if the price of an asset goes outside of a predetermined price band. The result is that if an asset experiences a huge drop in value (i.e. LUNA crash) the price of the oracle will continue to return the minPrice instead of the actual price of the asset. This would alLow user to continue borrowing with the asset but at the wrong price. This is exactly what happened to [Venus on BSC when LUNA imploded](https://rekt.news/venus-blizz-rekt/).	https://github.com/Fujicracy/fuji-v2/issues/333	chainlink oracle min value
27	Medium	2/25/2023 6:16	Attacker can withdraw contract's token balance	By taking funds stucked in the BaseRouter by depositing action and using different sender	https://github.com/Fujicracy/fuji-v2/issues/341	funds stuck in the router
30	Low	2/26/2023 19:17	Lack of validation to check whether or not the `REBALANCER_ROLE` and `LIQUIDATOR_ROLE` would be granted before the YieldVault/BorrowingVault is deployed	-	https://github.com/Fujicracy/fuji-v2/issues/346	concern with fuji roles
32	Low	2/27/2023 4:39	A user who has the `LIQUIDATOR_ROLE` will not lose their `gainedShares` even if their debt would be liquidated	A user who has the `LIQUIDATOR_ROLE` will not lose their `gainedShares` even if their debt would be liquidated	https://github.com/Fujicracy/fuji-v2/issues/347	concern with fuji roles
33	Low	2/27/2023 4:42	user's ETH can be lost while withdrawing	users eth assets can be lost while user is withdrawing their ETH using xBundle()	https://github.com/Fujicracy/fuji-v2/issues/349	base router withdrawETH checks
35	Reviewed	2/27/2023 5:42	Gas Optimizations		https://github.com/JustDravee/fuji-v2/compare/v0.0.1...JustDravee:fuji-v2:gas	Gas
36	Low	2/27/2023 6:05	previewWithdraw() should be Round up	Rounding of erc4626 in such way that in benefits the vault	https://github.com/Fujicracy/fuji-v2/issues/350	vault round-up round-down
37	Reviewed	2/27/2023 6:56	Gas Optimizations		https://github.com/Fujicracy/fuji-v2/compare/Fujicracy:fuji-v2:main...c3phas:fuji-v2:main	Gas
38	Low	2/27/2023 9:02	Missing deadline check in swap functions	Every swap function must have a deadline. Missing deadline checks alLow pending transactions to be maliciously executed in the future. You need to add a deadline parameter to all functions which potentially perform a swap on the user's behalf.	https://github.com/Fujicracy/fuji-v2/issues/351	router swap with deadline
39	Low	2/27/2023 9:08	Withdrawals are dependent on admin actions	A user can only use this function when whenNotPaused modifier is true. whenNotPaused modifier is controlled by the admin and so the admin can disable using the withdrawa(). User should not be restricted from using the withdraw function because of admin.	https://github.com/Fujicracy/fuji-v2/issues/352	vault implement share methods for debt

	Hats Finance Audit Competition Feb 14 - 27, 2023. Fuji Finance V2. See announcement					
#	Committee decision	Time (Arizona MST)	Submission title	Submission Description	GitHub issue Link	Grouped Categories
40	Low	2/27/2023 9:32	Unknown Repayment Amount	When a borrower repays an ERC20 loan, they can call the repayBorrow function with a specified amount to repay. However, interest accrues in every block, which means that if the borrower specifies the value of the loan at a particular block, their loan will be slightly higher in a future block when the transaction is confirmed. This could lead to the borrower leaving part of the loan unpaid.	https://github.com/Fujicracy/fuji-v2/issues/354	n/a
41	2nd Place Gas	2/27/2023 9:41	Gas Optimizations		https://github.com/Phantasmagoria13/fuji-v2/compare/v0.0.1...Phantasmagoria13:fuji-v2:main	Gas