DIVA Protocol V1 Audit

Overview

The audit was performed by six independent teams:

- Team 1: ComposableSecurity
- Team 2: gogo, kodyvim, Santipu_, zaskoh (SolidityLabs)
- Team 3: TrungOre, Duc (SolidityLabs)
- Team 4: said017, WangChao, kodak_rome, Emmalien (SolidityLabs)
- Team 5: devScrooge (JMariadlcs), Cryptor, Saksham (SolidityLabs)
- Team 6: HiAudit

The contracts in scope included:

- DIVA Protocol V1
- DIVA Development Fund
- Tellor oracle adapter
- DIVA Token distribution contract

> **Note:** The DIVA token distribution contract was not in scope for the SolidityLabs teams. It was audited by ComposableSecurity, HiAudit and an independent solo auditor.

> ⚠ **Warning:** HiAudit claims to be the #1 auditing firm in Japan but delivered a very disappointing report considering the amount of money we paid for their services. The team charged a flat fee rather than based on performance as advertised on ther website. The team appeared to lack sufficient experience and expertise in their field. Regrettably, we made a mistake in hiring them, and we strongly advise any other team against using their services. Surprisingly, they felt comfortable sharing their report publicly. Take your popcorn and search for "HiAudit".

Summary

The table below provides an overview of the findings grouped by contract. The numbers in parantheses indicate the resolved issues. The remaining findings were either acknowledged or declined. For details, please refer to the respective sections.

	Critical	High	Medium	Low	Informational	Gas optimization
DIVA Protocol	-	3 (3)	6 (5)	19 (11)	20 (9)	10 (5)
DIVA Development Fund	-	1 (1)	-	6 (2)	5 (3)	1 (1)
Tellor oracle adapter	-	-	-	1 (1)	6 (5)	-
DIVA Token distribution contract	-	-	-	-	3 (2)	-

DIVA Protocol

High

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
H-01	Wrong implementation of `EVMCall` in `DIVAOwnershipSecondary`		🔗					Resolved	Great unique finding! We'd like to highlight that if this error made it to production, the harm would be limited as Tellor reporters could have adopted the new query type. No user funds would have been at risk. Nonetheless, adhering to the proposed standard is the preferred approach.
H-02	Round-down calculation is used to calculate the `_collateralAmountRemovedNetMaker` which can be abused by taker to take all the removed liquidity from maker			🔗				Resolved	Great unique finding! While it wouldn't be economically viable to execute this attack, we agree to fix it to avoid any sort of griefing attack.
H-03	`_createContingentPoolLib` is suspicious of the reorg attack			🔗				Resolved (#29 / #48)	Very special and unique finding which helped us to better protect protocol users in the event of chain reorgs.

Medium

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
M-01	Wrong protocol fee recipient when withdrawing liquidity	🔗 (5.3)	🔗	🔗	🔗		🔗 (9)	Resolved	Good spot! We overlooked it when we updated the governance logic to introduce an activation delay. The impact would have been rather limited as the new treasury account would have received protocol fees two days earlier than expected. The purpose of the two-day delay for treasury updates was primarily to reduce the incentives for reporting incorrect owners on secondary chains. No user funds would have been at risk.

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
M-02	PreviousFallbackDataProvider won't have incentive to provide accurate value	(5.2)					(4)	Resolved	Same comment as for M-01, we overlooked it when we updated the governance logic. The impact would have been rather limited. We don't anticipate high TVL pools with a reputable data provider reaching a stage where the fallback data provider has to step in. In the unlikely event that such a scenario would have occured, the previous fallback provider could be incentivized through a direct payment to report the outcome. Additionally, it is worth noting that the fallback provider won't change too frequently.
M-03	Fee-on-Transfer tokens used as collateral will make a pool undercollateralized							Resolved	We agreed to block all fee-on-transfer tokens.
M-04	DoS in `_calcPayoffs` function when calculating big numbers							Resolved	Very special finding that no one else spotted!
M-05	`_getActualTakerFillableAmount` will return `_takerCollateralAmount - _offerInfo.takerFilledAmount` even if the order is not fillable							Resolved	Good finding that will help to avoid confusion for a certain class of offers. No user funds would have been at risk though.
M-06	Potentially In-Correct calculation of actual taker fillable amount						(3)	Declined	The HiAudit team failed to provide a more accurate formula than the existing one and refused to remove the issue from the report.

Low

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
L-01	Neither the long nor the short token can be conditionally burned							Resolved	
L-02	Trapped ETH in the Diamond contract							Resolved	
L-03	Missing important data in events							Resolved	
L-04	Don't allow setting owner to `address(0)` in `DIVAOwnershipSecondary`							Acknowledged	If a zero owner address was reported and remained undisputed, it would not result in any harm or negative consequences on secondary chains.

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
L-05	`DiamondCutFacet` should close the Diamond after getting called		🔗					Acknowledged	We decided to remove the upgradeability feature via a separate transaction rather than embedding it into the Diamond constructor to keep the code as close as possible to the original standard. In particular, if we ever plan to have an upgradeability feature in future versions of the protocol, we can achieve that without major code changes. Users will be able to verify that contracts are not upgradeable via https://louper.dev/, for instance.
L-06	Transferring a zero value amount may revert when creating a pool			🔗				Acknowledged	Not addressed as the amount > 0 check would be done within the corresponding ERC20 token.
L-07	Redundant requirement when requiring the `collateralAmount > 1e6` when creating a pool			🔗				Resolved	Great unique finding which helped us to reduce the gas cost for creating a contingent pool.
L-08	`unpauseReturnCollateral()` will extend pause delay time even when it already unpaused				🔗			Resolved	Great unique finding which helped us to improve the unpause functionality.
L-09	Griefer can challenge final reference value and prolonged the settlement process				🔗			Acknowledged	Not addressed as the possibility to confirm a previously submitted value by re-submitting the same value was a conscious design choice to prevent these type of attacks.
L-10	Centralization risk in token supply can result in users being unable to remove DIVA owner					🔗		Acknowledged	This issue is not a concern because power will eventually concentrate in the hands of those who have the highest belief in the project. Since these parties are likely to also stake for themselves, they will have a vested interest in acting in the best interest of the protocol.

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
L-11	Voting for a different owner can become impossible					🔗		Resolved	Great unique finding! The implemented solution to store the timestamp for “each stake” of each user would be an overkill. We have decided to store the staking timestamp at a user-candidate level instead of a user level as done before. This solves the problem if a user is staking for two different candidates. We acknowledge that the timestamp will be overwritten if a user stakes for the same candidate multiple times.
L-12	Diamond facet upgrade					🔗		Acknowledged	Not relevant as the protocol will be rendered immutable from the start.
L-13	Missing interface in IERC165	🔗 (5.4)						Resolved	
L-14	Unverified position token	🔗 (5.5)						Resolved	
L-15	Invalid receiver of settlement fee in liquidity removal	🔗 (5.7)						Resolved	Upon reviewing the recommendation, we discovered that our original (conscious) design choice could have led to incorrect settlement fee accounting within the Tellor adapter. To fix this issue, we applied a similar logic to the one used for tips, meaning that any accrued fees are held in a reserve and allocated to the corresponding recipient only after the final value has been confirmed.
L-16	Un-Satisfactory check while setting up `permissionedERC721Token`						🔗 (5)	Declined	The `permissionedERC721Token` address cannot be zero inside the `PermissionedPositionToken` contract as it's excluded in an `if` block inside the `PositionTokenFactory` contract. Despite highlighting this to HiAudit, they refused to remove this finding from the report, insisting that it aligns with best practices.

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
L-17	In-sufficient transfer check while allocating fees to `recipient`						(8)	Declined	Neither the treasury, the data provider nor the fallback provider can be the zero address (excluded inside the corresponding setter functions). Despite highlighting this to HiAudit, they refused to remove this finding from the report, insisting that it aligns with best practices.
L-18	Wrong implementation of EIP-2535 in LibDiamond library							Resolved	Resulted from using a slightly outdated version of the Diamond Standard which didn't include these optimizations.
L-19	Update openzeppelin NPM dependencies in package.json							Resolved	
L-20	Un-Satisfactory check while setting up owner						(1)	Declined	Despite pointing out to the HiAudit team that the owner of the position tokens is always the DIVA smart contract and can never be the zero address, they refused to remove this finding from the report, insisting that it aligns with best practices.

Informational

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-01	Missing function to query for `_permissionedPositionTokenImplementation` in `PositionTokenFactory`							Resolved	

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-02	Consider resetting values after a new Owner has claimed the ownership in `DIVAOwnershipMain`		🔗					Declined	Note that any non-winning candidate who has received more votes than the current owner can theoretically submit an ownership claim. That was a conscious design choice to simplify the snapshot logic. Resetting the values would allow a non-winning candidate to submit a claim and with that prevent the actual winner to submit their claim.
I-03	Misleading typo in comment		🔗					Resolved	

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-04	Violation Of Checks Effects Interation Pattern	(6.2)						Acknowledged	We have thoroughly evaluated the current implementation and are confident that it does not introduce any vulnerabilities. It was a conscious decision to prioritize drawing the capital before benefiting the `msg.sender`. Additionally, we have implemented reentrancy guards on all state-modifying functions (except governance related functions) to provide the necessary protection against reentrancy attacks.
I-05	Remove `poolId` from PoolStorage	(6.1)						Resolved	Resolved via H-03.
I-06	Improve code clarity	(6.3)						Resolved (PR6 / PR31)	Majority of the suggestions has been implemented.
I-07	Use proper error for non-existing pool	(6.4)						Resolved (#37 / #38 / #39 / #40 / #50)	

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-08	Add incentive for the default settlement	(6.5)						Declined	That was a conscious design choice. Position token holders will have a natural incentive to confirm the value and do not require additional incentives.
I-09	Optimize gas consumption by removing redundant checks	(6.6)						Resolved	
I-10	Avoid zero value transfers initiated by the protocol	(6.7)						Acknowledged	We believe that zero value transfers should be excluded on the frontend side rather than within the contract itself. Introducing the proposed check would result in additional gas costs. In particular, as we anticipate that data providers will utilize the `batchClaimFee` function, passing a collateral token with an amount of 0 by accident would cause the entire transaction to revert, leading to significant costs for the data provider.
I-11	Consider adding white hat hacks policy	(6.9)						Acknowledged	We will add a white hat hack policy at a later stage, post mainnet launch.

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-12	Consider extending the effect of the `pauseReturnCollateral` function	(6.12)						Declined	The decision to not implement the ability to pause the creation of derivative contracts was deliberate to prevent the owner from being pressured by a central authority to halt the entire protocol.
I-13	Add missing variable checks in constructor						(1)	Resolved	
I-14	Explicit Return [Code Readability]						(2)	Acknowledged	
I-15	Unclear usage when ERC20 blacklisted user removes liquidity						(6)	Acknowledged	A potential taker that gets blacklisted before filling a remove liquidity offer is equivalent to not having any taker at all. No user is losing any money in such a scenario. The maker can simply wait until expiry to redeem their funds. It doesn't need a taker to return the collateral. HiAudit's recommendation to implement a check to verify if a user is blacklisted is not realistic as any ERC20 token may implement a different function name.

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-16	Useless require statement at `_diamondCut` function		🔗					Resolved	Resulted from using a slightly outdated version of the Diamond Standard which didn't include these optimizations.
I-17	Missing NatSpec @inheritdoc in implementations		🔗					Acknowledged	If a function is not documented inside the implementation contract, then it's natural to check whether it's included in the interface. We don't see any value-add of adding the @inheritdoc NatSpec.
I-18	Missing NatSpec in diva-contracts Interfaces		🔗					Resolved	
I-19	Consider adding popups for front-end application to warn users	🔗 (6.11)						Acknowledged	This finding is frontend-related and not directly relevant for the smart contract itself.

Gas optimization

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
G-01	For Operations that will not overflow, you could use unchecked					🔗		Resolved	
G-02	Don't initialize variables with default value					🔗		Resolved	

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
G-03	Functions guaranteed to revert when called by normal users can be marked payable							Acknowledged	For the sake of consistency, we have made the decision to disallow the sending of ETH to the contract in any manner. Accidentally sending ETH to the contract could result in the loss of funds, which may outweigh any potential gas savings, especially, when considering that the mentioned governance functions are not anticipated to be utilized frequently. We acknowledge that our constructor is payable for gas optimization purposes, but this only affects the deployment process.
G-04	+i costs less gas than i++, especially when it's used in for-loops (--i/i-- too)							Resolved	
G-05	Use != 0 instead of > 0 for unsigned integer comparison							Resolved	
G-06	Internal functions only called once can be inlined							Acknowledged	We have chosen to leave it as is to prioritize code readability.
G-07	Using getter functions consume more gas							Acknowledged	We decided to leave it as is to avoid major code changes and the risk of introducing new bugs.
G-08	+= Costs More Gas							Declined	We somehow couldn't make the proposed syntax work as Remix flagged it as unsupported syntax.
G-09	ps Variable Can Be Inlined							Resolved	Very good one which helped to save some gas and a few lines of code.
G-10	Use while loop instead of for loop							Acknowledged	We decided to leave it as is to avoid major code changes and the risk of introducing new bugs.

Other

Issues not specifically raised by any of the auditing teams but related to other findings.

ID	Description	PR	Team comment
O-01	Remove outdated comments regarding upgradeability risk and owner right restrictions in Documentation	PR6	

DIVA Development Fund

High

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
H-01	Funds could be stuck in `DIVADevelopmentFund`	🔗 (5.1)	🔗					Resolved	

Low

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
L-01	Add a minimum deposit amount in `DIVADevelopmentFund`		🔗					Declined	Not addressed as someone could create a worthless token to circumvent such restriction.
L-02	Missing possibility of removing deposits that are fully paid in `DIVADevelopmentFund`		🔗					Declined	Not addressed as deleting array items would change the indices of deposits which is not desired. Also, the full array is never used, so we don't see any immediate benefit of deleting the items.
L-04	Missing important data in events		🔗					Resolved	
L-05	Fee-on-transfer tokens will get stuck in Development Fund	🔗 (5.6)	🔗					Resolved	
L-06	Missing validations while adding new deposit to address						🔗 (7)	Declined	Despite pointing out to the HiAudit team that the zero address does not implement the `safeTransferFrom` function, they refused to remove this finding from the report, insisting that it aligns with best practices.

Informational

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-01	Add missing variable checks in constructor		🔗					Resolved	
I-02	Improve code clarity	🔗 (6.3)						Resolved (#6 / #31)	Majority of the suggestions has been implemented.
I-03	Remove `payable` mutability from `withdraw` function	🔗 (6.10)						Acknowledged	We decided to leave it as is as the owner has the possibility to withdraw any directly deposited ETH.
I-04	Missing NatSpec @inheritdoc in implementations		🔗					Acknowledged	If a function is not documented inside the implementation contract, then it's natural to check whether it's included in the interface. We don't see any value-add of adding the @inheritdoc NatSpec.
I-05	Missing NatSpec in diva-contracts Interfaces		🔗					Resolved	

Gas optimization

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
G-01	Use custom error strings					🔗		Resolved	

Tellor oracle adapter

Low

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
L-01	Missing boundries for `_maxDIVARewardUSD` in `DIVAOracleTellor`		🔗					Acknowledged	As the purchasing power of USD may change over time, we agreed to not implement any boundaries.
L-02	Update openzeppelin NPM dependencies in package.json		🔗					Resolved	

Informational

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-01	Missing validation on deployment of DIVAOracleTellor		🔗					Resolved	
I-02	Use specific imports instead of just a global import in DIVAOracleTellor		🔗					Resolved	
I-03	Change immutable to constant if a fixed value is used		🔗					Resolved	
I-04	Add missing variable checks in constructor		🔗					Resolved	
I-05	Pragma version	🔗 (6.14)	🔗					Resolved	Decided to use Solidity version 0.8.19 for all contracts.
I-06	Missing NatSpec @inheritdoc in implementations		🔗					Acknowledged	If a function is not documented inside the implementation contract, then it's natural to check whether it's included in the interface. We don't see any value-add in adding the @inheritdoc NatSpec.

Other

Issues not specifically raised by any of the auditing teams but related to other findings.

	Description	PR	Team comment
O-01	Remove support for fee-on-transfer tokens in `addTip` function	#82	Related to finding M-03 finding in DIVA Protocol.
O-02	Update `poolId` type	#84	Necessary adjustment resulting from the new poolId logic implemented to protect against reorg attacks (see H-03 in DIVA Protocol).

	Description	PR	Team comment
O-03	Minor gas optimizations	#85	Related to the gas optimizations proposed for DIVA Protocol.

DIVA Token distribution contract

Informational

ID	Description	T1	T2	T3	T4	T5	T6	Status	Team comment
I-01	Detect duplicates in claimers' addresses	(6.8)						Resolved	
I-02	Protect withdrawing all tokens before setting up trigger	(6.13)						Acknowledged	Not addressed as this may be useful in case something goes wrong at initialization.
I-03	Pragma version	(6.14)						Resolved	

Other

Issues not specifically raised by any of the auditing teams but related to other findings.

ID	Description	PR	Team comment
O-01	Remove pause/unpause functionality from ClaimDIVALinearVesting contract	#13	We decided to remove the possibility to pause the contract to mitigate the risk of users having their tokens locked. Somewhat related to the centralization risk highlighted in DIVA Protocol (L-10).
O-02	Use custom errors instead of require to save users gas	#15	This issue is related to the gas optimization proposed for DIVA Development Fund.