METATRUST

Draft
Security Assessment for

43-LogicBug-Covalent (10K-WOI) (1Positive-FLP)

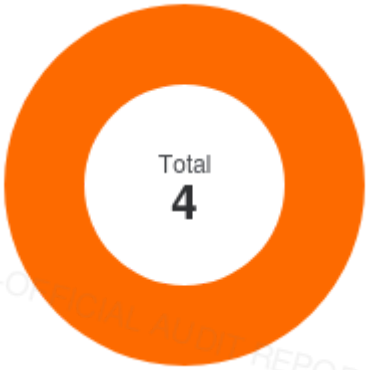
July 23, 2023

Executive Summary

Overview	
Project Name	43-LogicBug-Covalent (1OK-WOI) (1Positive-FLP)
Codebase URL	https://github.com/daoyuan14/LogicBug-Covalent
Scan Engine	AI Analyzer
Scan Time	2023/07/23 20:39:42
Commit Id	216a1b8

Total	
Critical Issues	0
High risk Issues	4
Medium risk Issues	0
Low risk Issues	0
Informational Issues	0

Critical Issues	The issue can cause large economic losses, large-scale data disorder, loss of control of authority management, failure of key functions, or indirectly affect the correct operation of other smart contracts interacting with it.
High Risk Issues	The issue puts a large number of users' sensitive information at risk or is reasonably likely to lead to catastrophic impacts on clients' reputations or serious financial implications for clients and users.
Medium Risk Issues	The issue puts a subset of users' sensitive information at risk, would be detrimental to the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk Issues	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational Issue	The issue does not pose an immediate risk but is relevant to security best practices or Defence in Depth.



Critical Issues	0%	0
High risk Issues	100%	4
Medium risk Issues	0%	0
Low risk Issues	0%	0
Informational Issues	0%	0

Summary of Findings

MetaScan security assessment was performed on **July 23, 2023 20:39:42** on project **43-LogicBug-Covalent (1OK-WOI) (1Positive-FLP)** with the repository **https://github.com/daoyuan14/LogicBug-Covalent** on branch **default branch**. The assessment was carried out by scanning the project's codebase using the scan engine **AI Analyzer**. There are in total **4** vulnerabilities / security risks discovered during the scanning session, among which **0** critical vulnerabilities, **4** high risk vulnerabilities, **0** medium risk vulnerabilities, **0** low risk vulnerabilities, **0** informational issues.

ID	Description	Severity
MSA-001	MWE-209: Wrong Order for Interest or ExchangeRate	High risk
MSA-002	MWE-200: Insecure LP Token Value Calculation	High risk
MSA-003	MWE-200: Insecure LP Token Value Calculation	High risk
MSA-004	MWE-209: Wrong Order for Interest or ExchangeRate	High risk

Findings

Critical (0)

No Critical vulnerabilities found here

High risk (4)

1. MWE-209: Wrong Order for Interest or ExchangeRate

High risk | Security Analyzer

Update of interest or exchange rate should be executed before calculating new balance, share, stake, loan or fee.

File(s) Affected

contracts/DelegatedStaking.sol #119-132

```
119     function updateGlobalExchangeRate() internal {
120         uint128 currentBlock = uint128(block.number);
121         // if the program ended, set update epoch to the end epoch
122         uint128 currentEpoch = currentBlock < endEpoch? currentBlock : endEpoch;
123         if (currentEpoch != lastUpdateEpoch){
124             // when no one has staked anything, do not update the rate
125             if(totalGlobalShares > 0)
126             {
127                 uint128 perEpochRateIncrease = uint128(uint256(allocatedTokensPerEpoch)*divider/uint256
128                 globalExchangeRate += perEpochRateIncrease * (currentEpoch - lastUpdateEpoch);
129             }
130             lastUpdateEpoch = currentEpoch;
131         }
132     }
```

Recommendation

Check the business logic and move the statements about updating exchange rate or interest forward.

2. MWE-200: Insecure LP Token Value Calculation

High risk Security Analyzer

Liquidity token value/price can be manipulated to cause flashloan attacks.

File(s) Affected

contracts/DelegatedStaking.sol #119-132

```
119     function updateGlobalExchangeRate() internal {
120         uint128 currentBlock = uint128(block.number);
121         // if the program ended, set update epoch to the end epoch
122         uint128 currentEpoch = currentBlock < endEpoch? currentBlock : endEpoch;
123         if (currentEpoch != lastUpdateEpoch){
124             // when no one has staked anything, do not update the rate
125             if(totalGlobalShares > 0)
126             {
127                 uint128 perEpochRateIncrease = uint128(uint256(allocatedTokensPerEpoch)*divider/uint25
128                 globalExchangeRate += perEpochRateIncrease * (currentEpoch - lastUpdateEpoch);
129             }
130             lastUpdateEpoch = currentEpoch;
131         }
132     }
```

contracts/DelegatedStaking.sol #308-346

```
308     function redeemRewards( uint128 validatorId, address beneficiary, uint128 amount) public {
309         require(beneficiary!=address(0x0), "Invalid beneficiary");
310         require(amount != 0, "Cannot redeem 0 tokens");
311         updateGlobalExchangeRate();
312         Validator storage v = validators[validatorId];
313         updateValidator(v);
314         Staking storage s = v.stakings[msg.sender];
315 
316         uint128 rewards = sharesToTokens(s.shares, v.exchangeRate) - s.staked;
317         if(msg.sender == v._address){
318             require(rewards + v.commissionAvailableToRedeem >= amount, "Cannot redeem more than availab
319             // first redeem rewards from commission
320             uint128 commissionLeftOver = amount < v.commissionAvailableToRedeem ? v.commissionAvailable
321             // if there is more, redeem  it from regular rewards
322             if (commissionLeftOver == 0){
323                 uint128 validatorSharesRemove = tokensToShares(amount - v.commissionAvailableToRedeem,
324                 s.shares -= validatorSharesRemove;
325                 v.totalShares -= validatorSharesRemove;
326             }
327             v.commissionAvailableToRedeem = commissionLeftOver;
328         }
329         else {
330             require(rewards >= amount, "Cannot redeem more than available");
331             uint128 validatorSharesRemove = tokensToShares(amount, v.exchangeRate);
332             s.shares -= validatorSharesRemove;
333             v.totalShares -= validatorSharesRemove;
334         }
335         transferFromContract(beneficiary, amount);
336 
337         // update global shares #
338         // this includes commission and rewards earned
339         // only update if the validator is enabled, otherwise the shares were already excluded during c
340         if (v.disabledEpoch == 0){
341             uint128 globalSharesRemove = tokensToShares(amount, globalExchangeRate);
342             totalGlobalShares -= globalSharesRemove;
343             v.globalShares -= globalSharesRemove;
344         }
345         emit RewardRedeemed(validatorId, beneficiary, amount);
346     }
```

Recommendation

Do not use AMM pool or custom liquidity calculation to caculate LP token value/price.

3. MWE-200: Insecure LP Token Value Calculation

High risk Security Analyzer

Liquidity token value/price can be manipulated to cause flashloan attacks.

File(s) Affected

contracts/DelegatedStaking.sol #135-157

```
135     function updateValidator(Validator storage v) internal {
136         // if validator is disabled, we do not update it since it was updated during disabling transact
137         if(v.disabledEpoch == 0){
138             if (v.totalShares == 0){
139                 // when validator stakes the first time, the exchange rate must be equal to the current
140                 v.exchangeRate = globalExchangeRate;
141             }
142             else {
143                 // the growth of global exchange rate since the validator was updated the last time
144                 uint128 rateDifference = globalExchangeRate - v.lastUpdateGlobalRate;
145                 // tokens given to the validator and its delegators since last update
146                 uint128 tokensGivenToValidator = sharesToTokens(v.globalShares, rateDifference);
147                 // commission paid out of the tokens
148                 uint128 commissionPaid = uint128(uint256(tokensGivenToValidator) * uint256(v.commissior
149                 // increase validator exchange rate by distributing the leftover tokens through the va
150                 v.exchangeRate += uint128(uint256(tokensGivenToValidator - commissionPaid) * divider /
151                 // give commission tokens to the validator
152                 v.commissionAvailableToRedeem += commissionPaid;
153             }
154             // set the last update global rate to the current one
155             v.lastUpdateGlobalRate = globalExchangeRate;
156         }
157     }
```

contracts/DelegatedStaking.sol #308-346

```
308     function redeemRewards( uint128 validatorId, address beneficiary, uint128 amount) public {
309         require(beneficiary!=address(0x0), "Invalid beneficiary");
310         require(amount != 0, "Cannot redeem 0 tokens");
311         updateGlobalExchangeRate();
312         Validator storage v = validators[validatorId];
313         updateValidator(v);
314         Staking storage s = v.stakings[msg.sender];
315 
316         uint128 rewards = sharesToTokens(s.shares, v.exchangeRate) - s.staked;
317         if(msg.sender == v._address){
318             require(rewards + v.commissionAvailableToRedeem >= amount, "Cannot redeem more than availab
319             // first redeem rewards from commission
320             uint128 commissionLeftOver = amount < v.commissionAvailableToRedeem ? v.commissionAvailable
321             // if there is more, redeem  it from regular rewards
322             if (commissionLeftOver == 0){
323                 uint128 validatorSharesRemove = tokensToShares(amount - v.commissionAvailableToRedeem,
324                 s.shares -= validatorSharesRemove;
325                 v.totalShares -= validatorSharesRemove;
326             }
327             v.commissionAvailableToRedeem = commissionLeftOver;
328         }
329         else {
330             require(rewards >= amount, "Cannot redeem more than available");
331             uint128 validatorSharesRemove = tokensToShares(amount, v.exchangeRate);
332             s.shares -= validatorSharesRemove;
333             v.totalShares -= validatorSharesRemove;
334         }
335         transferFromContract(beneficiary, amount);
336 
337         // update global shares #
338         // this includes commission and rewards earned
339         // only update if the validator is enabled, otherwise the shares were already excluded during 
340         if (v.disabledEpoch == 0){
341             uint128 globalSharesRemove = tokensToShares(amount, globalExchangeRate);
342             totalGlobalShares -= globalSharesRemove;
343             v.globalShares -= globalSharesRemove;
344         }
345         emit RewardRedeemed(validatorId, beneficiary, amount);
346     }
```

Recommendation

Do not use AMM pool or custom liquidity calculation to caculate LP token value/price.

4. MWE-209: Wrong Order for Interest or ExchangeRate

High risk Security Analyzer

Update of interest or exchange rate should be executed before calculating new balance, share, stake, loan or fee.

File(s) Affected

contracts/DelegatedStaking.sol #218-258

```
218     function unstake(uint128 validatorId, uint128 amount) public {
219         require(validatorId < validatorsN, "Invalid validator");
220         Validator storage v = validators[validatorId];
221         Staking storage s = v.stakings[msg.sender];
222         require(s.staked >= amount, "Staked is less than amount provided");
223         bool isValidator = msg.sender == v._address;
224         // only update if the validator is enabled, otherwise the global shares were already excluded
225         uint128 validatorSharesRemove = tokensToShares(amount, v.exchangeRate);
226         require(validatorSharesRemove > 0, "Unstake amount is too small");
227 
228         if (v.disabledEpoch == 0){
229             updateGlobalExchangeRate();
230             updateValidator(v);
231             // if validator is enabled and the program has not ended -> check for unstaking beyond max
232             if (isValidator && endEpoch > block.number){
233                 uint128 newValidatorStaked = s.staked - amount;
234                 uint128 newValidatorMaxCap = newValidatorStaked * maxCapMultiplier;
235                 uint128 delegated = v.delegated - s.staked;
236                 require(delegated <= newValidatorMaxCap, "Cannot unstake beyond max cap");
237                 require(newValidatorStaked >= validatorMinStakedRequired, "Cannot unstake beyond minimu
238             }
239 
240             // update global shares #
241             uint128 globalSharesRemove = tokensToShares(amount, globalExchangeRate);
242             require(globalSharesRemove > 0, "Unstake amount is too small");
243             totalGlobalShares -= globalSharesRemove;
244             v.globalShares -= globalSharesRemove;
245 
246             // update validator shares #
247             v.totalShares -= validatorSharesRemove;
248             v.delegated -= amount;
249         }
250         s.shares -= validatorSharesRemove;
251         s.staked -= amount;
252 
253         // create unstaking instance
254         uint128 coolDownEnd = v.disabledEpoch != 0 ? v.disabledEpoch : uint128(block.number);
255         coolDownEnd += (isValidator ? validatorCoolDown : delegatorCoolDown);
256         v.unstakings[msg.sender].push(Unstaking( coolDownEnd, amount));
257         emit Unstaked(validatorId, msg.sender, amount);
258     }
```

Recommendation

Check the business logic and move the statements about updating exchange rate or interest forward.

Medium risk (0)

No Medium risk vulnerabilities found here

Low risk (0)

No Low risk vulnerabilities found here

Informational (0)

No Informational vulnerabilities found here

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without MetaTrust's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts MetaTrust to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. MetaTrust's position is that each company and individual are responsible for their own due diligence and continuous security. MetaTrust's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by MetaTrust is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS 43-LogicBug-Covalent (1OK-WOI) (1Positive-FLP) Security Assessment AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW,

MetaTrust HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, MetaTrust SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, MetaTrust MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, MetaTrust PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER MetaTrust NOR ANY OF MetaTrust'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. MetaTrust WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT MetaTrust'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING 43-LogicBug-Covalent (1OK-WOI) (1Positive-FLP) Security Assessment MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST MetaTrust WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF MetaTrust CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST MetaTrust WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.