



METATRUST






Draft
Security Assessment for
25-2021-08-yield
(1New-FR) (10K-FLP)
(1Positive-FD)

July 23, 2023

Executive Summary

Overview			
Project Name	25-2021-08-yield (1New-FR) (1OK-FLP) (1Positive-FD)	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.
Codebase URL	https://github.com/code-423n4/2021-08-yield		
Scan Engine	AI Analyzer		
Scan Time	2023/07/23 01:13:36	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.
Commit Id	4dc4647		
Total		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.
Critical Issues	0		
High risk Issues	3	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.
Medium risk Issues	0		
Low risk Issues	0	Informational Issue	The issue does not pose an immediate risk but is relevant to security best practices or Defence in Depth.
Informational Issues	0		



	Critical Issues	0%	0
	High risk Issues	100%	3
	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 01:13:36** on project **25-2021-08-yield (1New-FR) (1OK-FLP) (1Positive-FD)** with the repository **<https://github.com/code-423n4/2021-08-yield>** 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 **3** vulnerabilities / security risks discovered during the scanning session, among which **0** critical vulnerabilities, **3** high risk vulnerabilities, **0** medium risk vulnerabilities, **0** low risk vulnerabilities, **0** informational issues.

ID	Description	Severity
MSA-001	MWE-204: Unsafe First Deposit	High risk
MSA-002	MWE-205: Front Running	High risk
MSA-003	MWE-200: Insecure LP Token Value Calculation	High risk


Findings


Critical (0)

No Critical vulnerabilities found here

High risk (3)

1. MWE-204: Unsafe First Deposit

 High risk

 Security Analyzer

First depositor can break minting of shares or drain the liquidity of all users.

File(s) Affected

contracts/yieldspace/Pool.sol #246-313

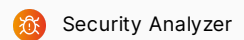
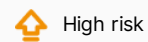
```
246     function _mintInternal(address to, bool calculateFromBase, uint256 fyTokenToBuy, uint256 minTokensMinted)
247         internal
248         returns (uint256, uint256, uint256)
249     {
250         // Gather data
251         uint256 supply = _totalSupply;
252         (uint112 _baseCached, uint112 _fyTokenCached) =
253             (baseCached, fyTokenCached);
254         uint256 _realFYTokenCached = _fyTokenCached - supply;    // The fyToken cache includes the virtual
255
256         // Calculate trade
257         uint256 tokensMinted;
258         uint256 baseIn;
259         uint256 baseReturned;
260         uint256 fyTokenIn;
261
262         if (supply == 0) {
263             require (calculateFromBase && fyTokenToBuy == 0, "Pool: Initialize only from base");
264             baseIn = base.balanceOf(address(this)) - _baseCached;
265             tokensMinted = baseIn;    // If supply == 0 we are initializing the pool and tokensMinted ==
266         } else {
267             // There is an optional virtual trade before the mint
268             uint256 baseToSell;
269             if (fyTokenToBuy > 0) {    // calculateFromBase == true and fyTokenToBuy > 0 can't happen
270                 baseToSell = _buyFYTokenPreview(
271                     fyTokenToBuy.u128(),
272                     _baseCached,
273                     _fyTokenCached
274                 );
275             }
276
277             if (calculateFromBase) {    // We use all the available base tokens, surplus is in fyTokens
278                 baseIn = base.balanceOf(address(this)) - _baseCached;
279                 tokensMinted = (supply * baseIn) / _baseCached;
280                 fyTokenIn = (_realFYTokenCached * tokensMinted) / supply;
281                 require(_realFYTokenCached + fyTokenIn <= fyToken.balanceOf(address(this)), "Pool: Not
282             } else {    // We use all the available fyTokens, plus a virtual trade if it
283                 fyTokenIn = fyToken.balanceOf(address(this)) - _realFYTokenCached;
284                 tokensMinted = (supply * (fyTokenToBuy + fyTokenIn)) / (_realFYTokenCached - fyTokenToBuy);
285                 baseIn = baseToSell + ((_baseCached + baseToSell) * tokensMinted) / supply;
286                 uint256 _baseBalance = base.balanceOf(address(this));
287                 require(_baseBalance - _baseCached >= baseIn, "Pool: Not enough base token in");
288
289                 // If we did a trade means we came in through `mintWithBase`, and want to return the base
290                 if (fyTokenToBuy > 0) baseReturned = (_baseBalance - _baseCached) - baseIn;
291             }
292         }
293
294         // Slippage
295         require (tokensMinted >= minTokensMinted, "Pool: Not enough tokens minted");
296
297         // Update TWAR
298         _update(
299             (_baseCached + baseIn).u128(),
300             (_fyTokenCached + fyTokenIn + tokensMinted).u128(), // Account for the "virtual" fyToken fi
301             _baseCached,
302             _fyTokenCached
```

```
303         );
304
305         // Execute mint
306         _mint(to, tokensMinted);
307
308         // Return any unused base if we did a trade, meaning slippage was involved.
309         if (supply > 0 && fyTokenToBuy > 0) base.safeTransfer(to, baseReturned);
310
311         emit Liquidity(maturity, msg.sender, to, -(baseIn.i256()), -(fyTokenIn.i256()), tokensMinted.i256());
312         return (baseIn, fyTokenIn, tokensMinted);
313     }
```

Recommendation

When totalSupply() == 0, send the first min liquidity LP tokens to the zero address to enable share dilution.

2. MWE-205: Front Running



Users are required to transfer assets in advance and minting token/liquidity/earning thus could be frontrun.

File(s) Affected



contracts/yieldspace/Strategy.sol #256-267

```
256     function mint(address to)
257     public
258     beforeMaturity
259     returns (uint256 minted)
260     {
261         // minted = supply * value(deposit) / value(strategy)
262         uint256 deposit = pool.balanceOf(address(this)) - cached;
263         minted = _totalSupply * deposit / cached;
264         cached += deposit;
265
266         _mint(to, minted);
267     }
```

Recommendation

Put asset transferring and token minting in the same function to keep atomicity.

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/oracles/compound/CTokenMultiOracle.sol #73-89

```
73     function _peek(bytes6 base, bytes6 quote) private view returns (uint price, uint updateTime){
74         uint256 rawPrice;
75         Source memory source = sources[base][quote];
76         require (source.source != address(0), "Source not found");
77
78         rawPrice = CTokenInterface(source.source).exchangeRateStored();
79
80         require(rawPrice > 0, "Compound price is zero");
81
82         if (source.inverse == true) {
83             price = 10 ** (source.decimals + 18) / uint(rawPrice);
84         } else {
85             price = uint(rawPrice) * 10 ** (18 - source.decimals);
86         }
87
88         updateTime = block.timestamp; // We should get the timestamp
89     }
```

contracts/oracles/composite/CompositeMultiOracle.sol #74-88

```
74     function peek(bytes32 base, bytes32 quote, uint256 amount)
75         external view virtual override
76         returns (uint256 value, uint256 updateTime)
77     {
78         uint256 price = 1e18;
79         bytes6 base_ = base.b6();
80         bytes6 quote_ = quote.b6();
81         bytes6[] memory path = paths[base_][quote_];
82         for (uint256 p = 0; p < path.length; p++) {
83             (price, updateTime) = _peek(base_, path[p], price, updateTime);
84             base_ = path[p];
85         }
86         (price, updateTime) = _peek(base_, quote_, price, updateTime);
87         value = price * amount / 1e18;
88     }
```

Recommendation

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

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 25-2021-08-yield (1New-FR) (1OK-FLP) (1Positive-FD) 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 25-2021-08-yield (1New-FR) (1OK-FLP) (1Positive-FD) 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.