



METATRUST

Draft
Security Assessment for
**61-2021-12-sublime
(1New-FD) (1Positive-
FLP)**

July 23, 2023






Executive Summary

Overview			
Project Name	61-2021-12-sublime (1New-FD) (1Positive-FLP)		
Codebase URL	https://github.com/code-423n4/2021-12-sublime		
Scan Engine	AI Analyzer		
Scan Time	2023/07/23 23:16:51		
Commit Id	9df1b7c		

Total			
Critical Issues	0		
High risk Issues	8		
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%	8
	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 23:16:51** on project **61-2021-12-sublime (1New-FD) (1Positive-FLP)** with the repository **<https://github.com/code-423n4/2021-12-sublime>** 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 **8** vulnerabilities / security risks discovered during the scanning session, among which **0** critical vulnerabilities, **8** high risk vulnerabilities, **0** medium risk vulnerabilities, **0** low risk vulnerabilities, **0** informational issues.

ID	Description	Severity
MSA-001	MWE-200: Insecure LP Token Value Calculation	High risk
MSA-002	MWE-204: Unsafe First Deposit	High risk
MSA-003	MWE-204: Unsafe First Deposit	High risk
MSA-004	MWE-204: Unsafe First Deposit	High risk
MSA-005	MWE-204: Unsafe First Deposit	High risk
MSA-006	MWE-200: Insecure LP Token Value Calculation	High risk
MSA-007	MWE-200: Insecure LP Token Value Calculation	High risk
MSA-008	MWE-200: Insecure LP Token Value Calculation	High risk


Findings

Critical (0)

No Critical vulnerabilities found here

High risk (8)

1. 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/mocks/yVault/Controller.sol #169-177

```
169     function getExpectedReturn(  
170         address _strategy,  
171         address _token,  
172         uint256 parts  
173     ) public view returns (uint256 expected) {  
174         uint256 _balance = IERC20(_token).balanceOf(_strategy);  
175         address _want = IStrategy(_strategy).want();  
176         (expected, ) = OneSplitAudit(onesplit).getExpectedReturn(_token, _want, _balance, parts, 0);  
177     }
```

Recommendation

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

2. 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/mocks/yVault/yVault.sol #81-87

```
81     function _mint(address account, uint256 amount) internal {
82         require(account != address(0), 'ERC20: mint to the zero address');
83
84         _totalSupply = _totalSupply.add(amount);
85         _balances[account] = _balances[account].add(amount);
86         emit Transfer(address(0), account, amount);
87     }
```



contracts/mocks/yVault/yVault.sol #275-288

```
275     function deposit(uint256 _amount) public {
276         uint256 _pool = balance();
277         uint256 _before = token.balanceOf(address(this));
278         token.safeTransferFrom(msg.sender, address(this), _amount);
279         uint256 _after = token.balanceOf(address(this));
280         _amount = _after.sub(_before); // Additional check for deflationary tokens
281         uint256 shares = 0;
282         if (totalSupply() == 0) {
283             shares = _amount;
284         } else {
285             shares = (_amount.mul(totalSupply()).div(_pool));
286         }
287         _mint(msg.sender, shares);
288     }
```

Recommendation

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

3. 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/mocks/yVault/yVault.sol #275-288

```
275     function deposit(uint256 _amount) public {
276         uint256 _pool = balance();
277         uint256 _before = token.balanceOf(address(this));
278         token.safeTransferFrom(msg.sender, address(this), _amount);
279         uint256 _after = token.balanceOf(address(this));
280         _amount = _after.sub(_before); // Additional check for deflationary tokens
281         uint256 shares = 0;
282         if (totalSupply() == 0) {
283             shares = _amount;
284         } else {
285             shares = (_amount.mul(totalSupply())).div(_pool);
286         }
287         _mint(msg.sender, shares);
288     }
```

Recommendation

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

4. 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/mocks/yVault/yVault.sol #275-288

```
275     function deposit(uint256 _amount) public {
276         uint256 _pool = balance();
277         uint256 _before = token.balanceOf(address(this));
278         token.safeTransferFrom(msg.sender, address(this), _amount);
279         uint256 _after = token.balanceOf(address(this));
280         _amount = _after.sub(_before); // Additional check for deflationary tokens
281         uint256 shares = 0;
282         if (totalSupply() == 0) {
283             shares = _amount;
284         } else {
285             shares = (_amount.mul(totalSupply()).div(_pool));
286         }
287         _mint(msg.sender, shares);
288     }
```



contracts/yield/AaveYield.sol #290-304

```
290     function _depositERC20(address asset, uint256 amount) internal returns (address aToken, uint256 sha
291         aToken = liquidityToken(asset);
292         uint256 aTokensBefore = IERC20(aToken).balanceOf(address(this));
293
294         address lendingPool = ILendingPoolAddressesProvider(lendingPoolAddressesProvider).getLendingPoc
295
296         //approve collateral to vault
297         IERC20(asset).approve(lendingPool, 0);
298         IERC20(asset).approve(lendingPool, amount);
299
300         //lock collateral in vault
301         AaveLendingPool(lendingPool).deposit(asset, amount, address(this), referralCode);
302
303         sharesReceived = IERC20(aToken).balanceOf(address(this)).sub(aTokensBefore);
304     }
```

Recommendation

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

5. 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/mocks/yVault/yVault.sol #290-304

```
290     function depositETH() public payable {
291         uint256 _pool = balance();
292         uint256 _before = token.balanceOf(address(this));
293         uint256 _amount = msg.value;
294         WETH(address(token)).deposit{value: _amount}();
295         uint256 _after = token.balanceOf(address(this));
296         _amount = _after.sub(_before); // Additional check for deflationary tokens
297         uint256 shares = 0;
298         if (totalSupply() == 0) {
299             shares = _amount;
300         } else {
301             shares = (_amount.mul(totalSupply()).div(_pool));
302         }
303         _mint(msg.sender, shares);
304     }
```

Recommendation

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

6. 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/Pool/Pool.sol #817-830

```
817     function _updateLenderSharesDuringLiquidation(address _lender)
818     internal
819     returns (uint256 _lenderCollateralLPShare, uint256 _lenderBalance)
820     {
821         uint256 _poolBaseLPShares = poolVariables.baseLiquidityShares;
822         _lenderBalance = balanceOf(_lender);
823
824         uint256 _lenderBaseLPShares = (_poolBaseLPShares.mul(_lenderBalance)).div(totalSupply());
825         uint256 _lenderExtraLPShares = lenders[_lender].extraLiquidityShares;
826         poolVariables.baseLiquidityShares = _poolBaseLPShares.sub(_lenderBaseLPShares);
827         poolVariables.extraLiquidityShares = poolVariables.extraLiquidityShares.sub(_lenderExtraLPShares);
828
829         _lenderCollateralLPShare = _lenderBaseLPShares.add(_lenderExtraLPShares);
830     }
```

contracts/Pool/Pool.sol #864-890

```
864     function liquidateForLender(
865         address _lender,
866         bool _fromSavingsAccount,
867         bool _toSavingsAccount,
868         bool _recieveLiquidityShare
869     ) external payable nonReentrant {
870         _canLenderBeLiquidated(_lender);
871
872         address _poolSavingsStrategy = poolConstants.poolSavingsStrategy;
873         (uint256 _lenderCollateralLPShare, uint256 _lenderBalance) = _updateLenderSharesDuringLiquidation(_lender);
874
875         uint256 _lenderCollateralTokens = _lenderCollateralLPShare;
876         _lenderCollateralTokens = IYield(_poolSavingsStrategy).getTokensForShares(_lenderCollateralLPShare);
877
878         _liquidateForLender(_fromSavingsAccount, _lender, _lenderCollateralTokens);
879
880         uint256 _amountReceived = _withdraw(
881             _toSavingsAccount,
882             _recieveLiquidityShare,
883             poolConstants.collateralAsset,
884             _poolSavingsStrategy,
885             _lenderCollateralTokens
886         );
887         _burn(_lender, _lenderBalance);
888         delete lenders[_lender];
889         emit LenderLiquidated(msg.sender, _lender, _amountReceived);
890     }
```

Recommendation

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

7. 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/yield/CompoundYield.sol #178-183

```
178     function getTokensForShares(uint256 shares, address asset) public override returns (uint256 amount)
179     //balanceOfUnderlying returns underlying balance for total shares
180     if (shares == 0) return 0;
181     address cToken = liquidityToken[asset];
182     amount = ICToken(cToken).balanceOfUnderlying(address(this)).mul(shares).div(IEC20(cToken).balanceOf(address(this)));
183 }
```

contracts/Pool/Pool.sol #864-890

```
864     function liquidateForLender(
865         address _lender,
866         bool _fromSavingsAccount,
867         bool _toSavingsAccount,
868         bool _recieveLiquidityShare
869     ) external payable nonReentrant {
870         _canLenderBeLiquidated(_lender);
871
872         address _poolSavingsStrategy = poolConstants.poolSavingsStrategy;
873         (uint256 _lenderCollateralLPShare, uint256 _lenderBalance) = _updateLenderSharesDuringLiquidation(_lender, _lenderBalance);
874
875         uint256 _lenderCollateralTokens = _lenderCollateralLPShare;
876         _lenderCollateralTokens = IYield(_poolSavingsStrategy).getTokensForShares(_lenderCollateralLPShare);
877
878         _liquidateForLender(_fromSavingsAccount, _lender, _lenderCollateralTokens);
879
880         uint256 _amountReceived = _withdraw(
881             _toSavingsAccount,
882             _recieveLiquidityShare,
883             poolConstants.collateralAsset,
884             _poolSavingsStrategy,
885             _lenderCollateralTokens
886         );
887         _burn(_lender, _lenderBalance);
888         delete lenders[_lender];
889         emit LenderLiquidated(msg.sender, _lender, _amountReceived);
890     }
```

Recommendation

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

Security Analyzer

File(s) Affected

```

256     function getTokensForShares(uint256 shares, address asset) public view override returns (uint256 ar
257         if (shares == 0) return 0;
258         address aToken = liquidityToken(asset);
259
260         ( , , , , , , , uint256 liquidityIndex, , ) = IProtocolDataProvider(protocolDataProvider).getRes
261
262         amount = IScaledBalanceToken(aToken).scaledBalanceOf(address(this)).mul(liquidityIndex).mul(sh
263             IERC20(aToken).balanceOf(address(this))
264     );
265 }

```

```

694     function calculateCollateralRatio(uint256 _balance, uint256 _liquidityShares) public returns (uint256)
695     {
696         uint256 _interest = interestToPay().mul(_balance).div(totalSupply());
697         address _collateralAsset = poolConstants.collateralAsset;
698         address _strategy = poolConstants.poolSavingsStrategy;
699         uint256 _currentCollateralTokens = IYield(_strategy).getTokensForShares(_liquidityShares, _collateralAsset);
700         uint256 _equivalentCollateral = getEquivalentTokens(_collateralAsset, poolConstants.borrowAsset);
701         _ratio = _equivalentCollateral.mul(10**30).div(_balance.add(_interest));
702     }

```

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

No Medium risk vulnerabilities found here

No Low risk vulnerabilities found here

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 61-2021-12-sublime (1New-FD) (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 61-2021-12-sublime (1New-FD) (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.