



METATRUST

Pre Report for






Derpdex-periphery

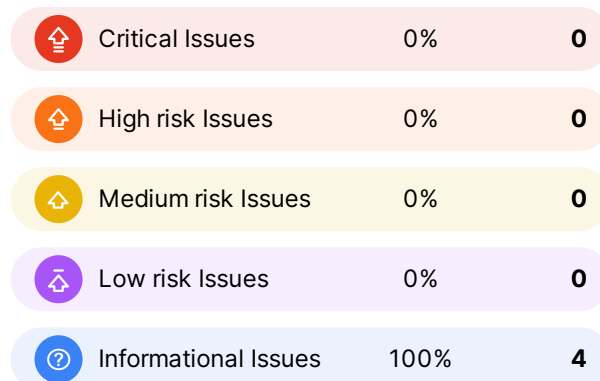
August 1, 2023

Executive Summary

Overview			
Project Name	Derpdex-periphery		
Codebase URL	https://github.com/derpdex-official/derpdex-periphery		
Scan Engine	Security Analyzer		
Scan Time	2023/08/1 17:06:10		
Commit Id	abdb2aa2		

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

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.



Summary of Findings

MetaScan security assessment was performed on **August 1, 2023 17:06:10** on project **Derpdex-periphery** with the repository **<https://github.com/derpdex-official/derpdex-periphery>** on branch **master**. The assessment was carried out by scanning the project's codebase using the scan engine **Security Analyzer**. There are in total **4** vulnerabilities / security risks discovered during the scanning session, among which **0** critical vulnerabilities, **0** high risk vulnerabilities, **0** medium risk vulnerabilities, **0** low risk vulnerabilities, **4** informational issues.

ID	Description	Severity
MSA-001	Unnecessary Public Function Modifier	Informational
MSA-002	Missing Zero Address Check	Informational
MSA-003	Unless POOL_INIT_CODE_HASH	Informational
MSA-004	Missing Error Message	Informational

Findings

Critical (0)

No Critical vulnerabilities found here

High risk (0)

No High risk vulnerabilities found here

Medium risk (0)


No Medium risk vulnerabilities found here


Low risk (0)

No Low risk vulnerabilities found here

Informational (4)

1. Unnecessary Public Function Modifier

 Informational

 Security Analyzer

There are some public functions that can be replaced as external functions.

File(s) Affected

contracts/base/Multicall.sol #11-27

```
11     function multicall(bytes[] calldata data) public payable override returns (bytes[] memory results) {
12         results = new bytes[] (data.length);
13         for (uint256 i = 0; i < data.length; i++) {
14             (bool success, bytes memory result) = address(this).delegatecall(data[i]);
15
16             if (!success) {
17                 // Next 5 lines from https://ethereum.stackexchange.com/a/83577
18                 if (result.length < 68) revert();
19                 assembly {
20                     result := add(result, 0x04)
21                 }
22                 revert(abi.decode(result, (string)));
23             }
24
25             results[i] = result;
26         }
27     }
```

contracts/base/PeripheryPayments.sol #30-41

```
30     function sweepToken(  
31         address token,  
32         uint256 amountMinimum,  
33         address recipient  
34     ) public payable override {  
35         uint256 balanceToken = IERC20(token).balanceOf(address(this));  
36         require(balanceToken >= amountMinimum, 'Insufficient token');  
37  
38         if (balanceToken > 0) {  
39             TransferHelper.safeTransfer(token, recipient, balanceToken);  
40         }  
41     }
```



contracts/base/PeripheryPayments.sol #19-27

```
19     function unwrapWETH9(uint256 amountMinimum, address recipient) public payable override {  
20         uint256 balanceWETH9 = IWETH9(WETH9).balanceOf(address(this));  
21         require(balanceWETH9 >= amountMinimum, 'Insufficient WETH9');  
22  
23         if (balanceWETH9 > 0) {  
24             IWETH9(WETH9).withdraw(balanceWETH9);  
25             TransferHelper.safeTransferETH(recipient, balanceWETH9);  
26         }  
27     }
```

Recommendation

Consider updating the public function as the external function for them.

2. Missing Zero Address Check

 Informational Security Analyzer

This function is lack of zero address check in important operation, which may cause some unexpected result.

File(s) Affected


contracts/NonfungibleTokenPositionDescriptor.sol #29-32

```
29     constructor(address _WETH9, bytes32 _nativeCurrencyLabelBytes) {  
30         WETH9 = _WETH9;  
31         nativeCurrencyLabelBytes = _nativeCurrencyLabelBytes;  
32     }
```

Recommendation

Consider adding zero address checks for address type storage variables.

3. Unless POOL_INIT_CODE_HASH

 Informational Security Analyzer

The `POOL_INIT_CODE_HASH` is not used by the `computeAddress` function and becomes unused for the project, thus, it is redundant. At the same time, its value of it is only right for Uniswap pool, and is wrong for DerpDex pool.

File(s) Affected

contracts/libraries/PoolAddress.sol #6-6

```
6     bytes32 internal constant POOL_INIT_CODE_HASH = 0xe34f199b19b2b4f47f68442619d555527d244f78a3297ea8932
```



contracts/libraries/PoolAddress.sol #33-43

```
33     function computeAddress(address factory, PoolKey memory key) internal view returns (address pool) {
34         require(key.token0 < key.token1);
35         (bool success, bytes memory addressBytes) = factory.staticcall(
36             abi.encodeWithSelector(
37                 bytes4(keccak256("getPool(address,address,uint24)")), key.token0, key.token1, key.fee)
38         );
39
40         require(success);
41         pool = abi.decode(addressBytes, (address));
42         require(pool != address(0));
43     }
```

Recommendation

Recommend removing the redundant constant variable `POOL_INIT_CODE_HASH`.

4. Missing Error Message

 Informational Security Analyzer

`require` can be used to check for conditions and throw an exception if the condition is not met, in which case the error message provided by the developer will appear. This is why a very descriptive error message is needed.

File(s) Affected

contracts/NonfungiblePositionManager.sol #316-316

```
316         require(params.amount0Max > 0 || params.amount1Max > 0);
```

contracts/NonfungiblePositionManager.sol #269-269

```
269         require(positionLiquidity >= params.liquidity);
```

contracts/libraries/PoolAddress.sol #34-34

```
34         require(key.token0 < key.token1);
```

contracts/SwapRouter.sol #62-62

```
62         require(amount0Delta > 0 || amount1Delta > 0); // swaps entirely within 0-liquidity regions are
```

Recommendation

Consider adding an error message describing all the failed condition.

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 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 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.