METATRUST

Security Assessment for

DerpDEX-Derp

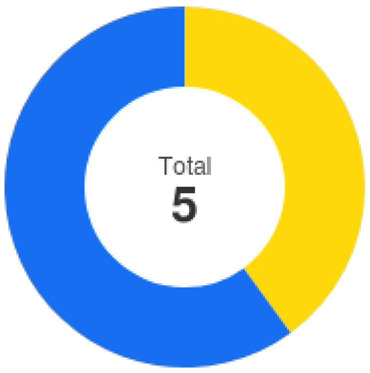
November 09, 2023

Executive Summary

Overview	
Project Name	DerpDEX-Derp
Codebase URL	https://github.com/derpdex-official/xDerp
Scan Engine	Security Analyzer
Scan Time	2023/11/09 08:00:00
Commit Id	e29aba95e7cc82adc5c616185dfaa16a4224d395 3df3b76728d22f91b9ee0a9e3d5f74ea7487c35b

Total	
Critical Issues	0
High risk Issues	0
Medium risk Issues	2
Low risk Issues	0
Informational Issues	3

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	0%	**0**
Medium risk Issues	40%	**2**
Low risk Issues	0%	**0**
Informational Issues	60%	**3**

Summary of Findings

MetaScan security assessment was performed on **November 09, 2023 08:00:00** on project **DerpDEX-Derp** with the repository on branch **default branch**. The assessment was carried out by scanning the project's codebase using the scan engine **Security Analyzer**. There are in total **5** vulnerabilities / security risks discovered during the scanning session, among which **0** critical vulnerabilities, **0** high risk vulnerabilities, **2** medium risk vulnerabilities, **0** low risk vulnerabilities, **3** informational issues.

ID	Description	Severity	Alleviation
MSA-001	Centralized Risk - Burn Token	Medium risk	Fixed
MSA-002	Centralized Risks	Medium risk	Acknowledged
MSA-003	Floating Pragma	Informational	Fixed
MSA-004	Lack of update the black list when adding a wallet into the black list	Informational	Fixed
MSA-005	Missing pragma statement	Informational	Fixed

Findings

Critical (0)

No Critical vulnerabilities found here

High risk (0)

No High risk vulnerabilities found here

Medium risk (2)

1. Centralized Risk - Burn Token

Medium risk Security Analyzer

In the `Derp` contract, the owner has the privilege of the following functions:

- `burn`: This function enables the owner to burn existing DERP tokens from a specified address.

The centralized function `burn` function is an extremely highly privileged function and looks unnecessary.

File(s) Affected

xDerp-master/contracts/Derp.sol #19-21

```
    function burn(address from, uint256 amount) external onlyOwner {
        _burn(from, amount);
    }
```

Recommendation

Recommend removing the privileged `burn` function.

Alleviation Fixed

The team replied that the team required this function as the team need to bridge the token for multiple chains while maintaining correct token supply. The idea the team attempt to do is mint new balance of the token on other chain, and burn the token supply on Ethereum mainnet.
Meanwhile, the team updated the `burn` function to only burn the owner's token in commit 1fe1afc993c08ce16cc0feee5f5d34a930abcbbe.

2. Centralized Risks

Medium risk Security Analyzer

In the `Derp` contract, the owner has the privilege of the following functions:

- `mint`: This function allows the owner to mint new DERP tokens and assign them to a specified address.
- `burn`: This function enables the owner to burn existing DERP tokens from a specified address.

In the `AntiSnipe` contract, the owner has the privilege of the following functions:

- `blacklist`: This function allows the owner to blacklist multiple addresses, preventing them from making transfers. It takes an array of target addresses as input and marks them as blacklisted.
- `updateWhitelists`: This function enables the owner to update the whitelists for multiple addresses at once. It takes two arrays as input: one for the target addresses and another for boolean values, allowing the owner to set or unset the whitelist status for each target.
- `updateWhitelist`: The owner can use this function to update the whitelist status for a single address. It takes an address and a boolean value as input, allowing the owner to add or remove the address from the whitelist.
- `enableTrading`: This function, when called by the owner, enables trading by setting the `canTrade` state variable to true, allowing transfers to occur. It checks if trading is already enabled and reverts if it is.
- `setAntiSnipeData`: This function allows the owner to set various anti-sniping parameters, including the pool contract address, the maximum allowed buy amount, the anti-snipe block interval, the maximum allowed balance, and whether selling is enabled. This function is used to configure anti-sniping measures.

File(s) Affected

xDerp-master/contracts/lib/AntiSnipe.sol #83-96

```
    function setAntiSnipeData(
        address _poolContract,
        uint256 _maxAllowedBuyAmount,
        uint256 _antiSnipeBlockInterval,
        uint256 _maxAllowedBalance,
        bool _sellEnabled
    ) external onlyOwner {
        poolContract = _poolContract;
        maxAllowedBuyAmount = _maxAllowedBuyAmount;
        antiSnipeBlockInterval = _antiSnipeBlockInterval;
        maxAllowedBalance = _maxAllowedBalance;
        antiSnipeStartBlock = block.number;
        sellEnabled = _sellEnabled;
    }
```

xDerp-master/contracts/lib/AntiSnipe.sol #60-64

```
    function blacklist(address[] calldata _targets) external onlyOwner {
        for(uint256 i=0; i< _targets.length; i++) {
            isBlacklisted[_targets[i]] = true;
        }
    }
```

xDerp-master/contracts/lib/AntiSnipe.sol #77-80

```
    function enableTrading() external onlyOwner {
        require(!canTrade, "ALREADY ENABLED");
        canTrade = true;
    }
```

xDerp-master/contracts/lib/AntiSnipe.sol #67-71

```
    function updateWhitelists(address[] calldata _targets, bool[] calldata value) external onlyOwner {
        for(uint256 i=0; i< _targets.length; i++) {
            _updateWhitelist(_targets[i], value[i]);
        }
    }
```

xDerp-master/contracts/Derp.sol #15-21

```
    function mint(address to, uint256 amount) external onlyOwner {
        _mint(to, amount);
    }

    function burn(address from, uint256 amount) external onlyOwner {
        _burn(from, amount);
    }
```

Recommendation

Consider implementing a decentralized governance mechanism or a multi-signature scheme that requires consensus among multiple parties before pausing or unpausing the contract. This can help mitigate the centralization risk associated with a single owner controlling critical contract functions. Alternatively, you can provide a clear justification for the centralization aspect and ensure that users are aware of the potential risks associated with a single point of control.

Alleviation Acknowledged

The team plans to move the ownership to DAO multisig wallet after listed on DEX in order to prevent bot sniping (https://derpdex.gitbook.io/home/derp-dao#derp-dao-framework).

Low risk (0)

No Low risk vulnerabilities found here

Informational (3)

1. Floating Pragma

Informational Security Analyzer

An unlocked compiler version like ^0.8.9 in the contract's source code permits the user to compile it at or above a particular version, which leads to differences in the generated bytecode between compilations due to differing compiler version numbers. As a result, compiler-specific bugs may occur in the codebase that would be hard to identify throughout multiple compiler versions rather than a specific one, which can cause ambiguity. Moreover, the contracts may be at the risk of being accidentally deployed using an outdated compiler version which can introduce bugs to affect the contract system negatively.

File(s) Affected

xDerp-master/contracts/Derp.sol #2-2

```
pragma solidity ^0.8.9;
```

Recommendation

Lock the compiler version to the lowest version possible so that the contract can be compiled and consider known bugs for the chosen compiler version.

Alleviation Fixed

The team resolved this issue in the commit 3df3b76728d22f91b9ee0a9e3d5f74ea7487c35b.

2. Lack of update the black list when adding a wallet into the black list

Informational Security Analyzer

There are two lists, **isWhitelisted** and **isBlacklisted**. Any wallet should be in at most one of the two lists.

To prevent a wallet in both of the above two lists. The **blacklist** function should exclude a wallet from the **isWhitelisted** list when adding the wallet to the **isBlacklisted**.

File(s) Affected

xDerp-master/contracts/lib/AntiSnipe.sol #60-64

```
    function blacklist(address[] calldata _targets) external onlyOwner {
        for(uint256 i=0; i< _targets.length; i++) {
            isBlacklisted[_targets[i]] = true;
        }
    }
```

Recommendation

Recommend excluding a wallet from the **isWhitelisted** list when adding the wallet to the **isBlacklisted** with the **blacklist** function.

Alleviation Fixed

The team resolved this issue in the commit 3df3b76728d22f91b9ee0a9e3d5f74ea7487c35b.

3. Missing pragma statement

Informational Security Analyzer

The **AntiSnipe** contract missing the pragma statement to specify the compiler version used to compile the smart contract.

Reference: The pragma keyword is used to enable certain compiler features or checks. A pragma directive is always local to a source file, so you have to add the pragma to all your files if you want to enable it in your whole project. If you import another file, the pragma from that file does not automatically apply to the importing file.

File(s) Affected

xDerp-master/contracts/lib/AntiSnipe.sol #1-1

```
// pragma solidity ^0.8.19;
```

Recommendation

Recommend adding a pragma statement to specify the compiler version.

Alleviation Fixed

The team resolved this issue in commit f454b498a43415378feb4fb592cc699d7f08f8ad.

Audit Scope

File	SHA256	File Path
AntiSnipe.sol	80579f51d34611663feb53391325f86df950d8eb5d0189 4f9bdd12ead986c949	/contracts/lib/AntiSnipe.sol
Derp.sol	84f193f4823f390553e74f81298332807431b1b0efcaff4 b46bff87c51116f9b	/contracts/Derp.sol

Disclaimer

This report is governed by the stipulations (including but not limited to service descriptions, confidentiality, disclaimers, and liability limitations) outlined in the Services Agreement, or as detailed in the scope of services and terms provided to you, the Customer or Company, within the context of the Agreement. The Company is permitted to use this report only as allowed under the terms of the Agreement. Without explicit written permission from MetaTrust, this report must not be shared, disclosed, referenced, or depended upon by any third parties, nor should copies be distributed to anyone other than the Company.

It is important to clarify that this report neither endorses nor disapproves any specific project or team. It should not be viewed as a reflection of the economic value or potential of any product or asset developed by teams or projects engaging MetaTrust for security evaluations. This report does not guarantee that the technology assessed is completely free of bugs, nor does it comment on the business practices, models, or legal compliance of the technology's creators.

This report is not intended to serve as investment advice or a tool for investment decisions related to any project. It represents a thorough assessment process aimed at enhancing code quality and mitigating risks inherent in cryptographic tokens and blockchain technology. Blockchain and cryptographic assets inherently carry ongoing risks. MetaTrust's role is to support companies and individuals in their security diligence and to reduce risks associated with the use of emerging and evolving technologies. However, MetaTrust does not guarantee the security or functionality of the technologies it evaluates.

MetaTrust's assessment services are contingent on various dependencies and are continuously evolving. Accessing or using these services, including reports and materials, is at your own risk, on an as-is and as-available basis. Cryptographic tokens are novel technologies with inherent technical risks and uncertainties. The assessment reports may contain inaccuracies, such as false positives or negatives, and unpredictable outcomes. The services may rely on multiple third-party layers.

All services, labels, assessment reports, work products, and other materials, or any results from their use, are provided "as is" and "as available," with all faults and defects, without any warranty. MetaTrust expressly disclaims all warranties, whether express, implied, statutory, or otherwise, including but not limited to warranties of merchantability, fitness for a particular purpose, title, non-infringement, and any warranties arising from course of dealing, usage, or trade practice. MetaTrust does not guarantee that the services, reports, or materials will meet specific requirements, be error-free, or be compatible with other software, systems, or services.

Neither MetaTrust nor its agents make any representations or warranties regarding the accuracy, reliability, or currency of any content provided through the services. MetaTrust is not liable for any content inaccuracies, personal injuries, property damages, or any loss resulting from the use of the services, reports, or materials.

Third-party materials are provided "as is," and any warranty concerning them is strictly between the Customer and the third-party owner or distributor. The services, reports, and materials are intended solely for the Customer and should not be relied upon by others or shared without MetaTrust's consent. No third party or representative thereof shall have any rights or claims against MetaTrust regarding these services, reports, or materials.

The provisions and warranties of MetaTrust in this agreement are exclusively for the Customer's benefit. No third party has any rights or claims against MetaTrust regarding these provisions or warranties. For clarity, the services, including any assessment reports or materials, should not be used as financial, tax, legal, regulatory, or other forms of advice.