

REPORT 60711303E39BAB001149B1F2

Created Sat Apr 10 2021 02:52:51 GMT+0000 (Coordinated Universal Time)  
Number of analyses 1  
User admin@aladdindefi.com

## REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
<a href="#">0b48280c-6fe1-45f6-b9f7-032eed50b815</a>	contracts/MasterChef.sol	50

Started	Sat Apr 10 2021 02:53:01 GMT+0000 (Coordinated Universal Time)
Finished	Sat Apr 10 2021 03:08:37 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Client Tool	Remythx
Main Source File	Contracts/MasterChef.Sol

## DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	23	27

## ISSUES

**MEDIUM** Function could be marked as external.

SWC-000 The function definition of "mint" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

## Source file

@openzeppelin/contracts/math/SafeMath.sol

## Locations

```
7  * checks.
8  *
9  * Arithmetic operations in Solidity wrap on overflow. This can easily result
10 * in bugs, because programmers usually assume that an overflow raises an
11 * error, which is the standard behavior in high level programming languages.
12 * 'SafeMath' restores this intuition by reverting the transaction when an
13 * operation overflows.
14 *
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "renounceOwnership" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

contracts/WishToken.sol

Locations

```
36 |
37 | /// @notice The EIP-712 typehash for the delegation struct used by the contract
38 | bytes32 public constant DELEGATION_TYPEHASH = keccak256("Delegation(address delegatee,uint256 nonce,uint256 expiry)");
39 |
40 | /// @notice A record of states for signing / validating signatures
41 | mapping (address => uint) public nonces;
42 |
43 | /// @notice An event thats emitted when an account changes its delegate
44 | event DelegateChanged(address indexed delegator, address indexed fromDelegate, address indexed toDelegate);
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "transferOwnership" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

contracts/WishToken.sol

Locations

```
42 |
43 | /// @notice An event thats emitted when an account changes its delegate
44 | event DelegateChanged(address indexed delegator, address indexed fromDelegate, address indexed toDelegate);
45 |
46 | /// @notice An event thats emitted when a delegate account's vote balance changes
47 | event DelegateVotesChanged(address indexed delegate, uint previousBalance, uint newBalance);
48 |
49 | /**
50 |  * @notice Delegate votes from `msg.sender` to `delegatee`
51 |  * @param delegator The address to get delegatee for
52 |  */
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "symbol" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
62 | * @dev Performs a Solidity function call using a low level `call`. A
63 | * plain `call` is an unsafe replacement for a function call: use this
64 | * function instead.
65 | *
66 | * If `target` reverts with a revert reason, it is bubbled up by this
67 | * function (like regular Solidity function calls).
68 | *
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "decimals" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
67 * function (like regular Solidity function calls).
68 *
69 * Returns the raw returned data. To convert to the expected return value.
70 * use https://solidity.readthedocs.io/en/latest/units-and-global-variables.html?highlight=abi.decode#abi-encoding-and-decoding-functions['abi.decode'].
71 *
72 * Requirements:
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "totalSupply" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
68 *
69 * Returns the raw returned data. To convert to the expected return value,
70 * use https://solidity.readthedocs.io/en/latest/units-and-global-variables.html?highlight=abi.decode#abi-encoding-and-decoding-functions['abi.decode'].
71 *
72 * Requirements
73 *
74 * - 'target' must be a contract.
75 * - calling 'target' with 'data' must not revert.
76 *
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "transfer" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
82
83 /**
84 * @dev Same as {xref-Address-functionCall-address-bytes-}[functionCall], but with
85 * 'errorMessage' as a fallback revert reason when 'target' reverts.
86 *
87 * Available since v3.1.
88 */
89 function functionCall(address target, bytes memory data, string memory errorMessage) internal returns (bytes memory) {
90     return functionCallWithValue(target, data, 0, errorMessage);
91 }
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "allowance" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
87 * _Available since v3.1._
88 */
89 function functionCall(address target, bytes memory data, string memory errorMessage) internal returns (bytes memory) {
90     return functionCallWithValue(target, data, 0, errorMessage);
91 }
92
93 /**
94  * @dev Same as {xref-Address-functionCall-address-bytes-}['functionCall'],
95  * but also transferring `value` wei to `target`.
96  *
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "approve" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
94 * @dev Same as {xref-Address-functionCall-address-bytes-}['functionCall'],
95 * but also transferring `value` wei to `target`.
96 *
97 * Requirements
98 *
99 * - the calling contract must have an ETH balance of at least `value`
100 * - the called Solidity function must be `payable`.
101 *
102 * _Available since v3.1._
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "transferFrom" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
108 | /**
109 |  * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[functionCallWithValue], but
110 |  * with 'errorMessage' as a fallback revert reason when 'target' reverts.
111 |  *
112 |  * _Available since v3.1.
113 |  */
114 | function functionCallWithValue(address target, bytes memory data, uint256 value, string memory errorMessage) internal returns (bytes memory) {
115 |     require(address(this).balance >= value, "Address: insufficient balance for call");
116 |     require(isContract(target), "Address: call to non-contract");
117 |
118 |     // solhint-disable-next-line avoid-low-level-calls
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "increaseAllowance" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
125 |  * but performing a static call.
126 |  *
127 |  * _Available since v3.3.
128 |  */
129 | function functionStaticCall(address target, bytes memory data) internal view returns (bytes memory) {
130 |     return functionStaticCall(target, data, "Address: low-level static call failed");
131 | }
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "decreaseAllowance" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
141 |
142 | // solhint-disable-next-line avoid-low-level-calls
143 | (bool success, bytes memory returndata) = target.staticcall(data);
144 | return _verifyCallResult(success, returndata, errorMessage);
145 |
146 |
147 | function _verifyCallResult(bool success, bytes memory returndata, string memory errorMessage) private pure returns bytes memory {
148 |     if (success) {
149 |         return returndata;
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "mint" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

@openzeppelin/contracts/utils/Address.sol

Locations

```
151 // Look for revert reason and bubble it up if present
152 if (returndata.length > 0) {
153 // The easiest way to bubble the revert reason is using memory via assembly
154
155 // solhint-disable-next-line no-inline-assembly
156 assembly {
157 let returndata_size := mload(returndata)
```