MythX

REPORT 60FB41EDFDA78800186953FC

Created	Fri Jul 23 2021 22:25:49 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	60fb400aa6e1846e55c6e980

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
1e65a520-0b70-485c-818b-726ef022eede	github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol	29

Analysis 1e65a520-0b70-485c-818b-726ef022eede MythX

Started	Fri Jul 23 2021 22:25:53 GMT+0000 (Coordinated Universal Time)
Finished	Fri Jul 23 2021 22:28:25 GMT+0000 (Coordinated Universal Time)
Mode	Quick
Client Tool	Remythx
Main Source File	Github/Coin-Of-Nature/Coin-Of-Nature/Coin_of_Nature.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	26	3

ISSUES

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
443 * thereby removing any functionality that is only available to the owner.
444 */
445 function renounceOwnership() public virtual onlyOwner {
446 emit OwnershipTransferred(_owner, address(0));
447 _owner = address(0);
448 }
449
450 /**
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
452 * Can only be called by the current owner.
453 */
454 function transferOwnership(address newOwner) public virtual onlyOwner {
455 require(newOwner != address(0), "Ownable: new owner is the zero address");
456 emit OwnershipTransferred(_owner, newOwner);
457 _owner = newOwner;
458 }
459
460 function geUnlockTime() public view returns (uint256) {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "geUnlockTime" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
458 }
459
460 function geUnlockTime() public view returns (uint256) {
461 return _lockTime;
462 }
463
464 //Locks the contract for owner for the amount of time provided
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "lock" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
463
464 //Locks the contract for owner for the amount of time provided
465 function lock(uint256 time) public virtual onlyOwner {
466 _previousOwner = _owner;
467 _owner = address(0);
468 _lockTime = now + time;
469 emit OwnershipTransferred(_owner, address(0));
470 }
471
472 //Unlocks the contract for owner when _lockTime is exceeds
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "unlock" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
471
472 //Unlocks the contract for owner when _lockTime is exceeds
473 function unlock() public virtual {
474 require(_previousOwner == msg.sender, "You don't have permission to unlock");
475 require(now > _lockTime , "Contract is locked until 7 days");
476 emit OwnershipTransferred(_owner, _previousOwner);
477 _owner = _previousOwner;
478 }
479 }
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "name" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
762 }
763
764 function name() public view returns (string memory) {
765 return _name;
766 }
767
768 function symbol() public view returns (string memory) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
766 }
767
768 function symbol() public view returns (string memory) {
769 return _symbol;
770 }
771
772 function decimals() public view returns (uint8) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
770 }
771
772 function decimals() public view returns (uint8) {
773 return _decimals;
774 }
775
776 function totalSupply() public view override returns (uint256) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
774 }
775
776 function totalSupply() public view override returns (uint256) {
777 return _tTotal;
778 }
779
780 function balanceOf(address account) public view override returns (uint256) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
783 }
784
785 function transfer(address recipient, uint256 amount) public override returns (bool) {
786 _transfer(_msgSender(), recipient, amount);
787 return true;
788 }
789
790 function allowance(address owner, address spender) public view override returns (uint256) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
788 }
789
790 function allowance(address owner, address spender) public view override returns (uint256) {
791 return _allowances[owner][spender];
792 }
793
794 function approve(address spender, uint256 amount) public override returns (bool) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
792 }
793
794 function approve(address spender, uint256 amount) public override returns (bool) {
795 _approve(_msgSender(), spender, amount);
796 return true;
797 }
798
799 function transferFrom(address sender, address recipient, uint256 amount) public override returns (bool) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
797 }
798
799 function transferFrom(address sender, address recipient, uint256 amount) public override returns (bool) {
800 _transfer(sender, recipient, amount);
801 _approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount, "ERC20: transfer amount exceeds allowance"));
802 return true;
803 }
804
805 function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
803 }
804
805 function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
806 _approve(_msgSender(), spender, _allowances[_msgSender()][spender].add(addedValue));
807 return true;
808 }
809
810 function decreaseAllowance(address spender, uint256 subtractedValue) public virtual returns (bool) {
```

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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
808 }
809
810 function decreaseAllowance(address spender, uint256 subtractedValue) public virtual returns (bool) {
811 _approve(_msgSender(), spender, _allowances[_msgSender()][spender].sub(subtractedValue, "ERC20: decreased allowance below zero"));
812 return true;
813 }
814
815 function isExcludedFromReward(address account) public view returns (bool) {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "isExcludedFromReward" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
813 }
814
815 function isExcludedFromReward(address account) public view returns (bool) {
816 return _isExcluded[account];
817 }
818
819 function totalFees() public view returns (uint256) {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "totalFees" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
817 }
818
819 function totalFees() public view returns (uint256) {
820 return _tFeeTotal;
821 }
822
823 function deliver(uint256 tAmount) public {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "deliver" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
821 }
822
823 function deliver(uint256 tAmount) public {
824 address sender = _msgSender();
825 require(!_isExcluded[sender], "Excluded addresses cannot call this function");
826 (uint256 rAmount,,,,,) = _getValues(tAmount);
827 _rOwned[sender] = _rOwned[sender].sub(rAmount);
828 _rTotal = _rTotal.sub(rAmount);
829 _tFeeTotal = _tFeeTotal.add(tAmount);
830 }
831
832 function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view returns(uint256) {
```

MEDIUM

SWC-000

Function could be marked as external.

The function definition of "reflectionFromToken" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
830 }
831
832 function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view returns(uint256) {
833 require(tAmount <= _tTotal, "Amount must be less than supply");
834 if (!deductTransferFee) {
835 (uint256 rAmount,,,,,) = _getValues(tAmount);
836 return rAmount;
837 } else {
838 (,uint256 rTransferAmount,,,,) = _getValues(tAmount);
839 return rTransferAmount;
840 }
841 }
842
843 function tokenFromReflection(uint256 rAmount) public view returns(uint256) {
```

MEDIUM

SWC-000

Function could be marked as external.

The function definition of "excludeFromReward" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
847 }
848
849 function excludeFromReward(address account) public onlyOwner() {
850 // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router.');
851 require(!_isExcluded[account], "Account is already excluded");
852 if(_rOwned[account] > 0) {
853 _tOwned[account] = tokenFromReflection(_rOwned[account]);
854 }
855 _isExcluded[account] = true;
856 _excluded.push(account);
857 }
858
859 function includeInReward(address account) external onlyOwner() {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "excludeFromFee" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
880 }
881
882 function excludeFromFee(address account) public onlyOwner {
883 _isExcludedFromFee[account] = true;
884 }
885
886 function includeInFee(address account) public onlyOwner {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "includeInFee" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
884 }
885
886 function includeInFee(address account) public onlyOwner {
887 _isExcludedFromFee[account] = false;
888 }
889
890 function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "setSwapAndLiquifyEnabled" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
902 }
903
904 function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
905 swapAndLiquifyEnabled = _enabled;
906 emit SwapAndLiquifyEnabledUpdated(_enabled);
907 }
908
909 //to recieve ETH from uniswapV2Router when swaping
```

MEDIUM

Function could be marked as external.

SWC-000

The function definition of "isExcludedFromFee" 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

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
988 }
989
990 function isExcludedFromFee(address account) public view returns(bool) {
991 return _isExcludedFromFee[account];
992 }
993
994 function _approve(address owner, address spender, uint256 amount) private {
```

MEDIUM

Read of persistent state following external call

SWC-107

The contract account state is accessed after an external call to a user defined address. To prevent reentrancy issues, consider accessing the state only before the call, especially if the callee is untrusted. Alternatively, a reentrancy lock can be used to prevent untrusted callees from re-entering the contract in an intermediate state.

Source file

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
756
757 //exclude owner and this contract from fee
758 _isExcludedFromFee[owner()] = true;
759 _isExcludedFromFee[address(this)] = true;
```

MEDIUM

Write to persistent state following external call

SWC-107

The contract account state is accessed after an external call to a user defined address. To prevent reentrancy issues, consider accessing the state only before the call, especially if the callee is untrusted. Alternatively, a reentrancy lock can be used to prevent untrusted callees from re-entering the contract in an intermediate state.

Source file

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
757 //exclude owner and this contract from fee
758 _isExcludedFromFee[owner()] = true;
759 _isExcludedFromFee[address(this)] = true;
760
761 emit Transfer(address(0), _msgSender(), _tTotal);
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.6.12"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
13 */
14
15 pragma solidity ^0.6.12;
16 // SPDX-License-Identifier: Unlicensed
17 interface IERC20 {
```

LOW

SWC-107

A call to a user-supplied address is executed.

An external message call to an address specified by the caller is executed. Note that the callee account might contain arbitrary code and could re-enter any function within this contract. Reentering the contract in an intermediate state may lead to unexpected behaviour. Make sure that no state modifications are executed after this call and/or reentrancy guards are in place.

Source file

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
749 IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
750 // Create a uniswap pair for this new token
751 uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
752 .createPair(address(this), _uniswapV2Router.WETH());
753
754 // set the rest of the contract variables
```

LOW

SWC-113

Multiple calls are executed in the same transaction.

This call is executed following another call within the same transaction. It is possible that the call never gets executed if a prior call fails permanently. This might be caused intentionally by a malicious callee. If possible, refactor the code such that each transaction only executes one external call or make sure that all callees can be trusted (i.e. they're part of your own codebase).

Source file

github/Coin-of-Nature/Coin-of-Nature/Coin_of_Nature.sol

Locations

```
750 // Create a uniswap pair for this new token
751 uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
752 .createPair(address(this), _uniswapV2Router.WETH());
753
754 // set the rest of the contract variables
```