MythX

REPORT 631A5B7A214FD50018448563

Created	Thu Sep 08 2022 21:15:38 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	62c4c7a5c8c2c74a2727f0ff

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
73bb029a-1ef3-4bdf-abe7-d29af60d2853	contracts/buycontract.sol	3

Analysis 73bb029a-1ef3-4bdf-abe7-d29af60d2853 MythX

Started	Thu Sep 08 2022 21:15:45 GMT+0000 (Coordinated Universal Time)
Finished	Thu Sep 08 2022 22:00:47 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	Contracts/Buycontract.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	3

ISSUES

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.7.0"". 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

contracts/buycontract.sol

Locations

```
1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity ^0.8.7.0;
4
5  import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721.sol";
```

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

contracts/buycontract.sol

Locations

```
10 openThetaContract(0xbB5f35D40132A0478f6aa91e79962e9F752167EA).createMarketSale{value:price}(NFTcontract, marketID);
11 ERC721 token = ERC721(NFTcontract);
12 token.transferFrom(address(this), customer, NFTid);
13 }
14 }
```

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

contracts/buycontract.sol

Locations

```
10  openThetaContract(0xbB5f35D40132A0478f6aa91e79962e9F752167EA).createMarketSale{value:price}(NFTcontract, marketID);
11  ERC721 token = ERC721(NFTcontract);
12  token.transferFrom(address(this), customer, NFTid);
13  }
14  }
```