MythX

REPORT 5EECC95ADC2FB6001147C05E

Created	Fri Jun 19 2020 14:19:06 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	yannik@chainsulting.de

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
4b064e05-8d39-4a02-b3e2-fa0bdf5b2157	browser/Nexxo_Smart_Contract_Solidity_2020.sol	8

Analysis 4b064e05-8d39-4a02-b3e2-fa0bdf5b2157

MythX

Started	Fri Jun 19 2020 14:19:17 GMT+0000 (Coordinated Universal Time)
Finished	Fri Jun 19 2020 15:04:28 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	Browser/Nexxo_Smart_Contract_Solidity_2020.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	8

ISSUES

LOW SWC-103

A floating pragma is set.

The current pragma Solidity directive is "">=0.5.3<=0.5.8"". 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

browser/Nexxo_Smart_Contract_Solidity_2020.sol

Locations

```
pragma solidity >=0.5.3 <=0.5.8;


```

LOW SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "_balances" is internal. Other possible visibility settings are public and private.

Source file

browser/Nexxo_Smart_Contract_Solidity_2020.sol

Locations

```
uint private INITIAL_SUPPLY;
uint private _totalSupply;
mapping(address => uint) _balances;

uint256 private _unitsOneEthCanBuy;
```

LOW

SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "_ownerWallet" is internal. Other possible visibility settings are public and private.

Source file

`browser/Nexxo_Smart_Contract_Solidity_2020.sol`

Locations

```
uint256 private _unitsOneEthCanBuy;
uint256 private _totalEthInWei;
address payable _ownerWallet;

mapping(address => mapping(address => uint)) _allowed;
```

LOW

SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "_allowed" is internal. Other possible visibility settings are public and private.

Source file

`browser/Nexxo_Smart_Contract_Solidity_2020.sol`

Locations

```
address payable _ownerWallet;

mapping(address => mapping(address => uint)) _allowed;

constructor(uint initalCapacity, uint256 unitsOneEthCanBuy, address payable ownerWallet) public {
```

LOW

SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "blockedAddressStructs" is internal. Other possible visibility settings are public and private.

Source file

`browser/Nexxo_Smart_Contract_Solidity_2020.sol`

Locations

```
}

mapping (address => EntityStruct) blockedAddressStructs;
address[] blockedAddressList; // unordered list of keys that actually exist
// End : to block and unblock particular address.
```

LOW

SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "blockedAddressList" is internal. Other possible visibility settings are public and private.

Source file

browser/Nexxo_Smart_Contract_Solidity_2020.sol

Locations

```

mapping (address => EntityStruct) blockedAddressStructs;
address[] blockedAddressList; // unordered list of keys that actually exist
// End : to block and unblock particular address.

```

LOW

SWC-128

Implicit loop over unbounded data structure.

Gas consumption in function "getBlockedAddressList" in contract "NexxoTokens" depends on the size of data structures that may grow unboundedly. The highlighted statement involves copying the array "blockedAddressList" from "storage" to "memory". When copying arrays from "storage" to "memory" the Solidity compiler emits an implicit loop.If the array grows too large, the gas required to execute the code will exceed the block gas limit, effectively causing a denial-of-service condition. Consider that an attacker might attempt to cause this condition on purpose.

Source file

browser/Nexxo_Smart_Contract_Solidity_2020.sol

Locations

```

function getBlockedAddressList() public onlyOwner view returns(address [] memory) {
return blockedAddressList;
}

```

LOW

SWC-134

Call with hardcoded gas amount.

The highlighted function call forwards a fixed amount of gas. This is discouraged as the gas cost of EVM instructions may change in the future, which could break this contract's assumptions. If this was done to prevent reentrancy attacks, consider alternative methods such as the checks-effects-interactions pattern or reentrancy locks instead.

Source file

browser/Nexxo_Smart_Contract_Solidity_2020.sol

Locations

```

emit Transfer(ownerWallet(), msg.sender, amount); // Broadcast a message to the blockchain
ownerWallet().transfer(msg.value); //Transfer ether to fundsWallet
}

```