MythX

REPORT 64E033BC21306D001AE19CD5

Created	Sat Aug 19 2023 03:15:08 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	648fc02af4bf584372592643

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
6fd98797-2e60-41b6-9c1a-602d8c5b42c6	/launchpad/zflaunchpadnative.sol	7

Analysis 6fd98797-2e60-41b6-9c1a-602d8c5b42c6 MythX

Started	Sat Aug 19 2023 03:15:14 GMT+0000 (Coordinated Universal Time)
Finished	Sat Aug 19 2023 04:00:24 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Vscode-Extension
Main Source File	/Launchpad/Zflaunchpadnative.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	7

ISSUES

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

/launchpad/zflaunchpadnative.sol

Locations

```
81 }
82
83 function withdrawFunds() external onlyOwner {
84 uint256 amount = address(this).balance;
85 _safeTransferETH(msg.sender, amount);
86 }
```

LOW

SWC-116

A control flow decision is made based on The block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/launchpad/zflaunchpadnative.sol

Locations

```
89 // No discount
90 if (block.timestamp < startTimestamp || block.timestamp.sub(startTimestamp).div(356400) > 0) { // No bonus
91 return 0;
92 }
93 uint256 _currentTime = block.timestamp.sub(startTimestamp);
94 // Zone 3
95 if (_currentTime >= 32400 && _currentTime.sub(32400).div(324000) == 0){
96 return 15 - (_currentTime.sub(32400)).div(21600);
```

LOW

SWC-116

A control flow decision is made based on The block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/launchpad/zflaunchpadnative.sol

Locations

```
89  // No discount
90  if (block.timestamp < startTimestamp || block.timestamp.sub(startTimestamp).div(356400) > 0) { // No bonus
91  return 0;
92  }
93  uint256 _currentTime = block.timestamp.sub(startTimestamp);
94  // Zone 3
95  if (_currentTime >= 32400 && _currentTime.sub(32400).div(324000) == 0){
96  return 15 - (_currentTime.sub(32400)).div(21600);
97  }
```

LOW

SWC-116

A control flow decision is made based on The block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/launchpad/zflaunchpadnative.sol

Locations

```
42  function deposit(address _referrer) payable external nonReentrant {
43  require(block.timestamp > startTimestamp, "deposit:Too early");
44  require(block.timestamp < endTimestamp, "deposit:Too late");
45  require(isSaleStart, "deposit: Sale not yet enabled");
46
47  uint256 _amount = msg.value;
```

LOW

SWC-116

A control flow decision is made based on The block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/launchpad/zflaunchpadnative.sol

Locations

```
44  require(block.timestamp < endTimestamp, "deposit:Too late");
45  require(isSaleStart, "deposit: Sale not yet enabled");
46
47  uint256 _amount = msg.value;
48
49  require(_amount > 0, "deposit:Amount must be > 0");
50
51  UserInfo storage user = userInfo[msg.sender];
```

LOW

SWC-116

A control flow decision is made based on The block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/launchpad/zflaunchpadnative.sol

Locations

```
94  // Zone 3
95  if (_currentTime >= 32400 && _currentTime.sub(32400).div(324000) == 0){
96  return 15 - (_currentTime.sub(32400)).div(21600);
97  }
98
99  return 25 - (_currentTime.div(10800)).mul(5) + (_currentTime.div(21600)).mul(5);
100 }
```

LOW

SWC-116

A control flow decision is made based on The block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/launchpad/zflaunchpadnative.sol

Locations

```
94  // Zone 3
95  if (_currentTime >= 32400 && _currentTime.sub(32400).div(324000) == 0){
96  return 15 - (_currentTime.sub(32400)).div(21600);
97  }
98
99  return 25 - (_currentTime.div(10800)).mul(5) + (_currentTime.div(21600)).mul(5);
100 }
101
102
103 function getUserInfo(address _user) public view returns (UserInfo memory) {
104 return userInfo[_user];
105 }
```