MythX

REPORT 6361D1031CFA66001A77E599

Created	Wed Nov 02 2022 02:08:03 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	62b1a8425ec4948f52c83856

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
70d0993e-22a7-44ca-933f-70c14852d230	contracts/BAPCv3.sol	3

Analysis 70d0993e-22a7-44ca-933f-70c14852d230 MythX

Started	Wed Nov 02 2022 02:08:08 GMT+0000 (Coordinated Universal Time)
Finished	Wed Nov 02 2022 02:23:59 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Client Tool	Remythx
Main Source File	Contracts/BAPCv3.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	3

ISSUES

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.4"". 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/erc721a/ERC721A.sol

Locations

```
23 *
24 * @dev Implementation of the [ERC721](https://eips.ethereum.org/EIPS/eip-721)
25 * Non-Fungible Token Standard, including the Metadata extension.
26 * Optimized for lower gas during batch mints.
27 *
```

LOW

SWC-107

Write to persistent state following external call.

The contract account state is accessed after an external call. 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

@openzeppelin/contracts/security/ReentrancyGuard.sol

Locations

```
59 // By storing the original value once again, a refund is triggered (see
60 // https://eips.ethereum.org/EIPS/eip-2200)
61 _status = _NOT_ENTERED;
62 }
63 }
```

LOW

SWC-115

Use of "tx.origin" as a part of authorization control.

The tx.origin environment variable has been found to influence a control flow decision. Note that using "tx.origin" as a security control might cause a situation where a user inadvertently authorizes a smart contract to perform an action on their behalf. It is recommended to use "msg.sender" instead.

Source file

contracts/erc721a/ERC721A.sol

Locations

```
51
52  // The bit position of `numberBurned` in packed address data.
53  uint256 private constant _BITPOS_NUMBER_BURNED = 128;
54
55  // The bit position of `aux` in packed address data.
```