MythX

REPORT 611FBE6AAB519A0018E7E9D8

Created	Fri Aug 20 2021 14:38:34 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	5e5c9d25627c330011a7b9be

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
8da23807-2a03-47f6-911d-40034bb87173	contracts/PolkaParty.sol	2

Analysis 8da23807-2a03-47f6-911d-40034bb87173 MythX

Started	Fri Aug 20 2021 14:38:39 GMT+0000 (Coordinated Universal Time)
Finished	Fri Aug 20 2021 15:23:45 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	`Contracts/PolkaParty.Sol`

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	2

ISSUES

UNKNOWN SWC-101 Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

`contracts/Interfaces/IERC20.sol`

Locations

```
* @dev Sets `amount` as the allowance of `spender` over the caller's tokens.
*
* Returns a boolean value indicating whether the operation succeeded.
*
* IMPORTANT: Beware that changing an allowance with this method brings the risk
```

UNKNOWN SWC-101 Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

`contracts/Interfaces/IERC20.sol`

Locations

```
* @dev Sets `amount` as the allowance of `spender` over the caller's tokens.
*
* Returns a boolean value indicating whether the operation succeeded.
*
* IMPORTANT: Beware that changing an allowance with this method brings the risk
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

`contracts/Interfaces/IERC20.sol`

Locations

```
* Returns a boolean value indicating whether the operation succeeded.
*
* IMPORTANT: Beware that changing an allowance with this method brings the risk
* that someone may use both the old and the new allowance by unfortunate
* transaction ordering. One possible solution to mitigate this race
```

UNKNOWN

SWC-101

Arithmetic operation "%" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

`contracts/Interfaces/IERC20.sol`

Locations

```
* Returns a boolean value indicating whether the operation succeeded.
*
* IMPORTANT: Beware that changing an allowance with this method brings the risk
* that someone may use both the old and the new allowance by unfortunate
* transaction ordering. One possible solution to mitigate this race
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

`contracts/Interfaces/IERC20.sol`

Locations

```
* Emits a {Transfer} event.
*/
function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);

/**
```

UNKNOWN

SWC-101

Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

contracts/Interfaces/IERC20.sol

Locations

```
* Emits a {Transfer} event.
*/
function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);

/**
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

contracts/Interfaces/IERC20.sol

Locations

```

/**
* @dev Emitted when `value` tokens are moved from one account (`from`) to
* another (`to`).
*
```

UNKNOWN

SWC-101

Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

contracts/Interfaces/IERC20.sol

Locations

```

/**
* @dev Emitted when `value` tokens are moved from one account (`from`) to
* another (`to`).
*
```

UNKNOWN

SWC-101

Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

contracts/Interfaces/IUniswapV2Pair.sol

Locations

```
function transferFrom(address from, address to, uint value) external returns (bool);

function DOMAIN_SEPARATOR() external view returns (bytes32);
function PERMIT_TYPEHASH() external pure returns (bytes32);
function nonces(address owner) external view returns (uint);
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

contracts/Interfaces/IUniswapV2Pair.sol

Locations

```
function permit(address owner, address spender, uint value, uint deadline, uint8 v, bytes32 r, bytes32 s) external;

event Mint(address indexed sender, uint amount0, uint amount1);
event Burn(address indexed sender, uint amount0, uint amount1, address indexed to);
event Swap(
address indexed sender,
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

contracts/Interfaces/IUniswapV2Pair.sol

Locations

```
function factory() external view returns (address);
function token0() external view returns (address);
function token1() external view returns (address);
function getReserves() external view returns (uint112 reserve0, uint112 reserve1, uint32 blockTimestampLast);
function price0CumulativeLast() external view returns (uint);
```

UNKNOWN

Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

contracts/Interfaces/IUniswapV2Pair.sol

Locations

```
function factory() external view returns (address);
function token0() external view returns (address);
function token1() external view returns (address);
function getReserves() external view returns (uint112 reserve0, uint112 reserve1, uint32 blockTimestampLast);
function price0CumulativeLast() external view returns (uint);
```

UNKNOWN

Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

contracts/Interfaces/IUniswapV2Pair.sol

Locations

```
function kLast() external view returns (uint);

function mint(address to) external returns (uint liquidity);
function burn(address to) external returns (uint amount0, uint amount1);
function swap(uint amount0Out, uint amount1Out, address to, bytes calldata data) external;
```

UNKNOWN

Arithmetic operation "%" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

contracts/Interfaces/IUniswapV2Pair.sol

Locations

```
function sync() external;

function initialize(address, address) external;
}
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is "">=0.7.6"". 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/Interfaces/IERC20.sol`

Locations

```

/**
* @dev Returns the amount of tokens owned by `account`.
*/
function balanceOf(address account) external view returns (uint256);
```

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 "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.

Source file

`contracts/Interfaces/IERC20.sol`

Locations

```
* allowance.
*
* Returns a boolean value indicating whether the operation succeeded.
*
* Emits a {Transfer} event.
*/
function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);
```