==== Dependence on predictable environment variable ====

SWC ID: 116

Severity: Low

Contract: AlphacoinCoin

Function name: transferFrom(address,address,uint256) or transferFrom(address,address,uint256)

PC address: 2066

Estimated Gas Usage: 2312 - 2977

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.

--------------------

In file: /home/mythril/APC.sol:96

balances[\_to] = balances[\_to].add(

--------------------

Initial State:

Account: [CREATOR], balance: 0xa03, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x8, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [ATTACKER], function: transferFrom(address,address,uint256), txdata: 0x23b872dd000000000000000000000000008000000000000000000000000000000000000000000000000000000000000010, value: 0x0

==== Dependence on predictable environment variable ====

SWC ID: 116

Severity: Low

Contract: AlphacoinCoin

Function name: transfer(address,uint256) or transfer(address,uint256)

PC address: 3808

Estimated Gas Usage: 1396 - 1681

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.

--------------------

In file: /home/mythril/APC.sol:80

end);

// SafeMath.sub will throw if there is not e

--------------------

Initial State:

Account: [CREATOR], balance: 0x212, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x8, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [ATTACKER], function: transfer(address,uint256), txdata: 0xa9059cbb00000000000000000000000001, value: 0x0

==== Dependence on predictable environment variable ====

SWC ID: 116

Severity: Low

Contract: AlphacoinCoin

Function name: transfer(address,uint256) or transfer(address,uint256)

PC address: 3895

Estimated Gas Usage: 1413 - 1698

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.

--------------------

In file: /home/mythril/APC.sol:80

end);

// SafeMath.sub will throw if there is not enough balance.

balances

--------------------

Initial State:

Account: [CREATOR], balance: 0x40000002000000408, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [CREATOR], function: transfer(address,uint256), txdata: 0xa9059cbb00000000000000000000000001, value: 0x0

==== Dependence on predictable environment variable ====

SWC ID: 116

Severity: Low

Contract: AlphacoinCoin

Function name: transfer(address,uint256) or transfer(address,uint256)

PC address: 3984

Estimated Gas Usage: 1433 - 1718

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.

--------------------

In file: /home/mythril/APC.sol:80

= multisend);

// SafeMath.sub will throw if there is not enough balance.

balances[

--------------------

Initial State:

Account: [CREATOR], balance: 0x40080002002020080, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [SOMEGUY], function: transfer(address,uint256), txdata: 0xa9059cbb00000000000000000000000001, value: 0x0

==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: AlphacoinCoin

Function name: increaseApproval(address,uint256)

PC address: 5085

Estimated Gas Usage: 1502 - 1977

An assertion violation was triggered.

It is possible to trigger an assertion violation. Note that Solidity assert() statements should only be used to check invariants. Review the transaction trace generated for this issue and either make sure your program logic is correct, or use require() instead of assert() if your goal is to constrain user inputs or enforce preconditions. Remember to validate inputs from both callers (for instance, via passed arguments) and callees (for instance, via return values).

--------------------

In file: /home/mythril/APC.sol:110

\_spender]);

return true;

}

function d

--------------------

Initial State:

Account: [CREATOR], balance: 0x4204000400a00440f, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x2000000008000409, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [ATTACKER], function: approve(address,uint256), txdata: 0x095ea7b3000000000000000000000000aaaaaaaaaaaaaaaaaaaaaa8aaa2aaa000000000005, value: 0x0

Caller: [ATTACKER], function: increaseApproval(address,uint256), txdata: 0xd73dd623060606060606060606060606aaaaaaaaaaaaaaaaaaaaaa8aaa2aaa0000000000fb, value: 0x0