==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: BasicToken

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

PC address: 826

Estimated Gas Usage: 1219 - 1504

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/LINK.sol:88

Transfer(msg.sender, \_to, \_value

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

Initial State:

Account: [CREATOR], balance: 0x7bbb9, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x1, nonce:0, storage:{}

Transaction Sequence:

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

Caller: [ATTACKER], function: transfer(address,uint256), txdata: 0xa9059cbb000000000000000000000000000000000000000000000000000000000000002001, value: 0x0

==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: StandardToken

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

PC address: 3666

Estimated Gas Usage: 1285 - 1570

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/LINK.sol:88

Transfer(msg.sender, \_to, \_value

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

Initial State:

Account: [CREATOR], balance: 0x2000000040004001, nonce:0, storage:{}

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

Transaction Sequence:

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

Caller: [CREATOR], function: transfer(address,uint256), txdata: 0xa9059cbb000000000000000000000000000000000000000000000000000000000000000008, value: 0x0

==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: StandardToken

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

PC address: 3666

Estimated Gas Usage: 2243 - 2908

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/LINK.sol:130

lue);

Transfer(\_from, \_

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

Initial State:

Account: [CREATOR], balance: 0x1000019140b3, nonce:0, storage:{}

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

Transaction Sequence:

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

Caller: [SOMEGUY], function: transferFrom(address,address,uint256), txdata: 0x23b872dd0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000080, value: 0x0

==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: LinkToken

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

PC address: 4752

Estimated Gas Usage: 1489 - 1774

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/LINK.sol:88

Transfer(msg.sender, \_to, \_value

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

Initial State:

Account: [CREATOR], balance: 0x2100000100008103, nonce:0, storage:{}

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

Transaction Sequence:

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

Caller: [SOMEGUY], function: transfer(address,uint256), txdata: 0xa9059cbb000000000000000000000000000000000000000000000000000000808040040004, value: 0x0

==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: LinkToken

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

PC address: 4752

Estimated Gas Usage: 2384 - 3049

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/LINK.sol:130

lue);

Transfer(\_from, \_

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

Initial State:

Account: [CREATOR], balance: 0x80, nonce:0, storage:{}

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

Transaction Sequence:

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

Caller: [CREATOR], function: transferFrom(address,address,uint256), txdata: 0x23b872dd000000000000000000000000affeaffeaffeaffeaffeaffeaffeaffeaffeaffe000000000000000000000000000000000000000000000000000000000002000001, value: 0x0

==== Exception State ====

SWC ID: 110

Severity: Medium

Contract: LinkToken

Function name: transferAndCall(address,uint256,bytes) or transferAndCall(address,uint256,bytes) or transferAndCall(address,uint256,bytes)

PC address: 4752

Estimated Gas Usage: 1647 - 4519

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/LINK.sol:88

Transfer(msg.sender, \_to, \_value

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

Initial State:

Account: [CREATOR], balance: 0x1, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x201, nonce:0, storage:{}

Transaction Sequence:

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

Caller: [ATTACKER], function: transferAndCall(address,uint256,bytes), txdata: 0x4000aea0000000000000000000000000deadbeefdeadbeefdeadbeefdeadbeefdeadbeef01, value: 0x0