Quantstamp Audit Results

Requestor Address: 0x6B9aaa08aeA41eE4cED230cff1cc23e6710006f1

Auditor Address: 0x9bf5e4620Db5944BCCf0A1E6a6b72A430c5A0126

Contract URI: https://s3.amazonaws.com/qsp-protocol-test-contracts/abc9341dcce-b5e3-418b-9c00-167e0df73eaa.sol

Contract Hash: 90b555f2ef5327e1896001d87462e2e69da3daace18cb7b60f22312493a9c4f9

Vulnerability	**Oyente**	**Mythril**
integer_underflow	Success	Success
integer_overflow	Success	Success
callstack	Success	Not Tested
money_concurrency	Success	Not Tested
time_dependency	Success	Not Tested
reentrancy	Failure	Not Tested
parity_multisig_bug_2	Success	Not Tested
assertion_failure	Success	Not Tested
call_data_forwarded	Not Tested	Success
dependence_on_environment_variable	Not Tested	Success
call_to_a_user-supplied_address	Not Tested	Success
use_of_tx_origin	Not Tested	Success
ether_send	Not Tested	Failure
exception_state	Not Tested	Success
message_call_to_external_contract	Not Tested	Failure
state_change_after_external_call	Not Tested	Failure
multiple_calls	Not Tested	Success
unchecked_suicide	Not Tested	Success
transaction_order_dependence	Not Tested	Success
unchecked_call_return_value	Not Tested	Success

More details about potential vulnerabilities:

ReportName: oyente, name: reentrancy, type: undefined, file: abc9341dcce-b5e3-418b-9c00-167e0df73eaa.sol, contract: SendBalance, description: undefined, on start line: 15

ReportName: mythril, name: Ether send, type: ether_send, file: abc9341dcce-b5e3-418b-9c00-167e0df73eaa.sol, contract: undefined, description: In the function `withdrawBalance()` a non-zero amount of

Ether is sent to msg.sender. It seems that this function can be called without restrictions., on start line: 15

ReportName: mythril, name: Message call to external contract, type: message_call_to_external_contract, file: abc9341dcce-b5e3-418b-9c00-167e0df73eaa.sol, contract: undefined, description: This contract executes a message call to the address of the transaction sender. Generally, it is not recommended to call user-supplied addresses using Solidity's call() construct. Note that attackers might leverage reentrancy attacks to exploit race conditions or manipulate this contract's state., on start line: 15

ReportName: mythril, name: State change after external call, type: state_change_after_external_call, file: abc9341dcce-b5e3-418b-9c00-167e0df73eaa.sol, contract: undefined, description: The contract account state is changed after an external call. Consider that the called contract could re-enter the function before this state change takes place. This can lead to business logic vulnerabilities., on start line: 19