### SWC ID : 113

pragma solidity 0.4.24;

contract Refunder {

address[] private refundAddresses;mapping (address => uint) public refunds;

constructor() {

refundAddresses.push(0x79B483371E87d664cd39491b5F06250165e4b184);

refundAddresses.push(0x79B483371E87d664cd39491b5F06250165e4b185);

}

// bad

function refundAll() public {

for(uint x; x < refundAddresses.length; x++) { // arbitrary length iteration based on how many addresses participated

require(refundAddresses[x].send(refunds[refundAddresses[x]])); // doubly bad, now a single failure on send will hold up all funds

}

}

}

**SWC ID : 128**

pragma solidity ^0.4.25;

contract DosGas {

address[] creditorAddresses;

bool win = false;

function emptyCreditors() public {

if(creditorAddresses.length>1500) {

creditorAddresses = new address[](0);

win = true;

}

}

function addCreditors() public returns (bool) {

for(uint i=0;i<350;i++) {

creditorAddresses.push(msg.sender);

}

return true;

}

function iWin() public view returns (bool) {

return win;

}

function numberCreditors() public view returns (uint) {

return creditorAddresses.length;

}

}

pragma solidity ^0.4.25;

contract DosNumber {

uint numElements = 0;

uint[] array;

function insertNnumbers(uint value,uint numbers) public {

// Gas DOS if number > 382 more or less, it depends on actual gas limit

for(uint i=0;i<numbers;i++) {

if(numElements == array.length) {

array.length += 1;

}

array[numElements++] = value;

}

}

function clear() public {

require(numElements>1500);

numElements = 0;

}

// Gas DOS clear

function clearDOS() public {

// number depends on actual gas limit

require(numElements>1500);

array = new uint[](0);

numElements = 0;

}

function getLengthArray() public view returns(uint) {

return numElements;

}

function getRealLengthArray() public view returns(uint) {

return array.length;

}

}

pragma solidity ^0.4.25;

contract DosOneFunc {

address[] listAddresses;

function ifillArray() public returns (bool){

if(listAddresses.length<1500) {

for(uint i=0;i<350;i++) {

listAddresses.push(msg.sender);

}

return true;

} else {

listAddresses = new address[](0);

return false;

}

}

}

**SWC ID : 134**

pragma solidity 0.6.4;

interface ICallable {

function callMe() external;

}

contract HardcodedNotGood {

address payable \_callable = 0xaAaAaAaaAaAaAaaAaAAAAAAAAaaaAaAaAaaAaaAa;

ICallable callable = ICallable(\_callable);

constructor() public payable {

}

function doTransfer(uint256 amount) public {

\_callable.transfer(amount);

}

function doSend(uint256 amount) public {

\_callable.send(amount);

}

function callLowLevel() public {

\_callable.call.value(0).gas(10000)("");

}

function callWithArgs() public {

callable.callMe{gas: 10000}();

}

}