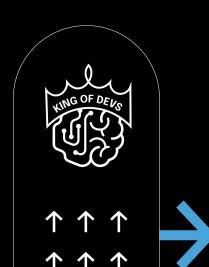
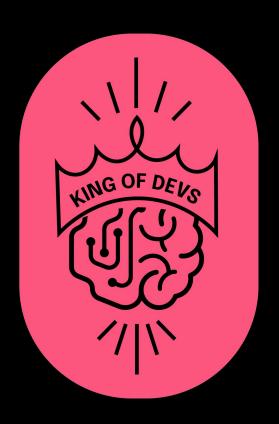
think & dev

KING OF DEVS





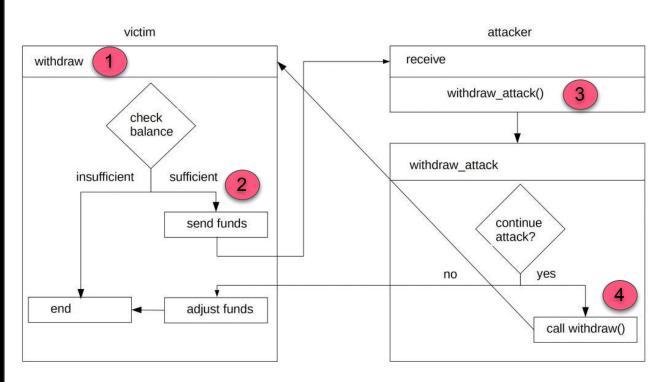
WORKSHOP DAY 2

RE-ENTRANCY





QUE ES UN RE-ENTRANCY ATTACK?



```
pragma solidity ^0.8.9;
 4 ∨ contract ReentrancyVictim {
         mapping(address => uint256) public balances;
         //Declare events
         event Deposit(address indexed from, uint256 value);
         event Withdraw(address indexed _from, uint256 _value);
         // Allows the caller to add balance
         function getBalance(address _address) public view returns (uint256 balance) {
             balance = balances[_address];
15
         // Allows the caller to add balance
          function payIn() public payable {
             emit Deposit(msg.sender, msg.value);
             balances[msg.sender] += msg.value;
         // Withdraws the balance available to the caller
         function withdraw() public payable {
             require(balances[msg.sender] > 0, "Insufficient balance");
             emit Withdraw(msg.sender, balances[msg.sender]);
             (bool success, ) = payable(msg.sender).call{value: balances[msg.sender]}("");
             require(success, "Failed to send funds");
             balances[msg.sender] = 0;
          function contractBalance() public view returns (uint256) {
             return address(this).balance;
```

□Unsafe code example

□Execution logs

```
1 ----------Deploying reentrancy example (ReentrancyVictim)------
2 Deployer user address: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
3 RentrancyVictimInstance deployed to: 0x5FbDB2315678afecb367f032d93F642f64180aa3
4 RentrancyAttackerInstance with 1 ETH deployed to: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512
5 ----------Deployed reentrancy example contracts--------
6 Event Deposit. From: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266. Value: 9
7 Event Deposit. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
8 Victim contract total balance before attack: 10
9 Attacker contract total balance before attack: 0
10 Proceed with attack? (Y,n)
11 Performing attack
12 Event Withdraw. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
13 Event Withdraw. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
14 Event Withdraw. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
15 Event Withdraw. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
16 Victim contract total balance after attack: 6
17 Attacker contract total balance after attack: 4
```

KING OF DEVS

contracts/development/reentrancy/re 1 vulnerable.sol

think & dev think & dev

```
contract ReentrancyProtected {
    event Deposit(address indexed from, uint256 value);
    event Withdraw(address indexed _from, uint256 _value);
    function getBalance(address _address) public view returns (uint256 balance) {
        balance = balances[ address];
    function payIn() public payable {
    function withdraw() public payable {
        uint256 balance = balances[msg.sender];
       balances[msg.sender] = 0;
        (bool success, ) = payable(msg.sender).call{value: balance}("");
        require(success, "Failed to send funds");
        emit Withdraw(msg.sender, balance);
    function contractBalance() public view returns (uint256) {
        return address(this).balance;
```

Safe code example

□Execution logs

```
1 -------Deploying reentrancy example (ReentrancyProtected)------
2 Deployer user address: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
3 RentrancyVictimInstance deployed to: 0x5FbDB2315678afecb367f032d93F642f64180aa3
4 RentrancyAttackerInstance with 1 ETH deployed to: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512
5 --------Deployed reentrancy example contracts------
6 Event Deposit. From: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266. Value: 9
7 Event Deposit. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
8 Victim contract total balance before attack: 10
9 Attacker contract total balance before attack: 0
10 Proceed with attack? (Y,n)
11 Performing attack
12 Event Withdraw. From: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512. Value: 1
13 Victim contract total balance after attack: 9
14 Attacker contract total balance after attack: 1
```

KING OF DEVS

contracts/development/reentrancy/re 2 final.sol

think & dev think & dev

```
contract ReentrancyWithOZ is ReentrancyGuard {
    mapping(address => uint256) public balances;
    event Deposit(address indexed from, uint256 value);
    event Withdraw(address indexed _from, uint256 _value);
    function getBalance(address _address) public view returns (uint256 balance) {
        balance = balances[_address];
        emit Deposit(msg.sender, msg.value);
    function withdraw() public payable nonReentrant {
        emit Withdraw(msg.sender, balances[msg.sender]);
        (bool success, ) = payable(msg.sender).call{value: balances[msg.sender]}("");
        return address(this).balance:
```

□Safe code (with OZ) example

 $\underline{\text{https://qithub.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/securit}}_{v/ReentrancvGuard.sol}$

□Execution logs

AΤ

BLOCK HOUSE

contracts/development/reentrancy/re 3 oz modifier.sol

think & dev think & dev





GRACIAS KINNG OF DELICATION OF

