dasp top 10

1. Reentrancy (재진입)

**function** **withdraw**(uint \_amount) {

require(balances[msg.sender] >= \_amount);

msg.sender.call.value(\_amount)();

balances[msg.sender] -= \_amount;

}

(Msg.sender.call.value() 는 무슨의미일까?)

Reentrancy가 발생하면 무한으로 발생하는지 아니면 1회만 작동할거 3회 작동한다던지?

예시

Withdraw()함수로 외부주소의 잔고를 추적하고 자금을 인출할수있음.

악의적인 계약은 withdraw()를 사용하여 전체 잔고를 인출하려고함.

피해를 받는 계약은 악의적인 계약에게 이더를전송하기위해

call.value(amount)() 라는 low level 함수를 악의적인 계약의 잔고를업데이트 하기 전에 사용함

악의적인 계약은 payable fallback() 함수로 자금을 받고 피해받는계약의 withdraw()함수를 다시 호출한다.

두번째 실행은 자금의 전송을 일으킴. 그러나 악의적인 계약의 잔고는 첫번째 인출에서 업데이트 되지 않음. 결과적으로 악의적인 계약은 두번째에서 모든 잔고를 인출해감.

2.Access control

3.Arithmetic

4.Unchecked Low Level Calls

5.Denial of Services

6.Bad Randomness

7.Front Running

8.Time Manipulation

9.Short Addresses

10. Unknown Unknowns