Web3 Security Researcher:

Rosario Borgesi
Blockchain Engineer

collectFee & reentrancy: Transfer the owner’s balance (not contract balance) and update it first using the CEI pattern.
Safety & simplification: Replace all transfer with call and avoid unnecessary uint256 < 0 checks.

✕ BUG

Vault.sol

```
function withdraw(uint value) external {
   if (payment[msg.sender] <= value) revert("insufficient balance");
         // ✕ BUG-2
         payable(msg.sender).transfer(value);
}

function collectFee() external onlyOwner {
         if (payment[msg.sender] <= 0) revert("insufficient balance");
         // ✕ BUG-1,2
         payable(msg.sender).transfer(address(this).balance);
   }
```

✓ FIX

Vault.sol

```
function withdraw(uint value) external nonReentrant {
         // ✓ FIX-2
         (bool success, ) = payable(msg.sender).call{value: value}("");
         if (!success) revert TransferFailed();
```

```
    }

    function collectFee() external onlyOwner {
        uint256 amount = payment[msg.sender];
        if (amount == 0) revert insufficientBalance();
        // ✓ FIX-1,2
        payment[msg.sender] = 0;
        (bool success, ) = payable(msg.sender).call{value: amount}("");
        if (!success) revert TransferFailed();
        emit feeCollected(amount, msg.sender);
    }
```

🛡️ FIX: Selected key fixes from several important suggestions