Nome: Joao Otavio Rodrigues de Castro Manieri

Matrícula: 12021BSI263

Bug 1 - Transferência de Token Antes do Leilão

Problema:

No método `createAuction`, o leilão é configurado, mas o token pode ser transferido antes de o leilão começar. Isso pode resultar na perda do token se o leilão não for iniciado.

Consequências:

O proprietário original perde o controle do token sem garantia de que o leilão será concluído, o que pode levar a problemas de segurança e confiabilidade.

Nova Solução:

Em vez de transferir o token no `initAuction`, podemos armazenar o token como referência e transferi-lo apenas se o leilão chegar à fase final, eliminando a necessidade de movimentação de token no início.

Código Corrigido:

```
function createAuction(string memory name, uint time, VerySimpleToken t)
public {

    require(t.isOwner(msg.sender), "You must own the token to create an auction!");

    OneAuction memory l;

    l.blocklimit = block.number + time;

    l.myState = AuctionStates.Prep;

    l.tokenOwner = msg.sender;

    l.winnerBid = 0;
```

```
    l.token = t;

    myAuctions[name] = l;

}
```

No final, no método `claimToken`, adicionamos a transferência do token caso o leilão seja concluído:

```
function claimToken(string memory name) public payable {

    OneAuction storage a = myAuctions[name];

    verifyFinished(a);

    require(a.myState == AuctionStates.Finished, "Auction is not finished yet");

    require(a.winner == msg.sender, "Only the winner can claim the token");

    require(msg.value == a.winnerBid - collateralValue, "Complete payment
required");


    // Transfer token only when the auction is truly finished

    a.token.transfer(msg.sender);

}
```

Bug 2 - Liberação de Colateral Indevida

Problema:

O vencedor do leilão tem o colateral liberado ao usar a função `claimToken`, o que não deveria acontecer, pois pode permitir manipulação indesejada no processo.

Consequências:

Permite que o vencedor libere o colateral incorretamente, podendo causar falhas na integridade financeira do contrato.

Nova Solução:

Adicionar uma função separada para liberar colateral apenas para aqueles que não venceram, garantindo que o colateral do vencedor seja tratado independentemente.

Código Corrigido:

function releaseWinnerCollateral(string memory name) private {

OneAuction storage a = myAuctions[name];

a.collateral[a.winner] = false; // Liberação do colateral apenas para o vencedor, separadamente

}

Bug 3 - Retirada Total no getFee

Problema:

A função `getFee` retira todo o saldo do contrato, incluindo fundos necessários para cobrir colaterais e lances vencedores, o que pode causar inadimplência.

Consequências:

Pode esgotar o saldo do contrato, impossibilitando a devolução de colaterais e o pagamento de lances.

Nova Solução:

Em vez de calcular o saldo disponível, podemos mover as taxas automaticamente para uma carteira separada conforme os lances e taxas são recebidos. Isso simplifica o controle e garante que fundos críticos permaneçam no contrato.

address payable feeAccount; // Conta para armazenar as taxas de maneira separada

Código Corrigido:

```
constructor(uint c, uint fee, address payable _feeAccount) public {
    owner = msg.sender;
    collateralValue = c;
    contractFee = fee;
    feeAccount = _feeAccount;
}

// Quando o pagamento é feito, uma porcentagem é
automaticamente transferida para `feeAccount`
function transferFee(uint amount) private {
    uint fee = amount * contractFee / 1000;
    feeAccount.transfer(fee);
}
```