Lab 7 – Vulnerabilidades

El contrato CryptoVault[1] (fichero `CryptoVault.sol`) implementa un servicio para almacenar Ether de clientes. Este servicio cobra una comisión que se resta de las cantidades depositadas cuando se invoca a la función `deposit`. Esta comisión se calcula utilizando un porcentaje que se fija cuando se crea el contrato. Del mismo modo, el propietario del contrato puede recoger las comisiones obtenidas hasta el momento invocando la función `collectFees`.

Este contrato tiene varias vulnerabilidades similares a las que hemos visto en clase (al menos tres).

1. Identifica esas vulnerabilidades, detallando la(s) función(es) y las líneas de código que las producen.

2. Por cada vulnerabilidad detectada, crea un contrato malicioso para atacar `CryptoVault`. Cada contrato debe tener un constructor que reciba como parámetro la dirección del contrato `CyptoVault` a atacar y una función `attack` para realizar el ataque. Esta función puede ser **`payable`** si el ataque necesita Ether para ser realizado. También puedes añadir una función pública `getBalance` que devuelva el saldo del contrato malicioso para comprobar que el ataque funciona correctamente.

3. Por cada vulnerabilidad, proporciona la secuencia de operaciones que deben ejecutarse para realizar el ataque, detallando las cuentas de usuario involucradas, las cantidades de Ether transferidas y los resultados obtenidos.

4. Propón una versión mejorada de `CryptoVault` que corrija todas las vulnerabilidades detectadas. Por cada línea modificada, especifica con un comentario a su derecha la vulnerabilidad que se corrige con ese cambio. También puedes añadir comentarios adicionales explicando los cambios realizados. Utiliza los siguientes tipos de corrección:

   - Si has detectado una vulnerabilida de reentrada, utiliza un mecanismo basado en *locks* para corregirla.
   - Si has detectado una vulnerabilidad de *overflow/underflow*, **no cambies la versión del compilador:** modifica la expresión aritmética o utiliza `SafeMath` para realizar operaciones aritméticas seguras.

5. Por cada vulnerabilidad detectada, explica por qué los ataques descritos en los apartados 2 y 3 no pueden tener éxito con los cambios realizados a `CryptoVault`.

[1] Este contrato no tiene ninguna relación con el producto comercial https://cryptovault.net/