**Akropolis Protocol**

The Akropolis Protocol is another contract that users can deposit funds and earn interest. The protocol deposit function was vulnerable to a reentrancy attack.

The Akropolis Protocol deposit method gets as arguments the tokens types to deposit and the amounts, and performs as follows:

1. Record total balance worth at the beginning of the transaction.
2. Transfer the funds safely – if the user doesn’t have the funds then reverts.
3. Calculate total balance worth and the difference from step 1.
4. Mint tokens corresponding to the added tokens worth calculated at step 3.

The attack is deposit([FAKE], [1]).

* FAKE token is implemented such that transferFrom(user1, user2, amount) call deposit with 1M DAI tokens.
* What happens is that step 1 records value .  
  Then in step 2 after applying transferFrom of FAKE the value increased by and the attacker gets 1M minted Akropolis tokens due to the deposit.  
  Then at step 3 the calculated difference is also 1M, therefore another 1M minted Akropolis tokens are transferred to the attaker.
* The attacker ends with 2M tokens which worth 2M DAI while transferred only 1M DAI.  
  Then the attacker can withdraw its tokens and get the DAI.

Here the reentrancy isn’t with the fallback function but with the transferFrom function. To fix that, the Akropolis Protocol should support deposit of trusted tokens only, which their transferFrom function is implemented safely.