Writing Secured Smart Contracts Quiz

Neri must build the ultimate fortress to stop Hackana! Test your knowledge of smart contract security best practices.

---

1. What is the "Checks-Effects-Interactions" pattern used for?

A) To make contracts run faster

B) To prevent reentrancy attacks by updating state before external calls

C) To allow anyone to call any function

D) To increase gas costs

Answer: B

Explanation: This pattern helps prevent reentrancy attacks by updating contract state before making external calls.

---

2. What does the `nonReentrant` modifier from OpenZeppelin do?

A) Allows multiple calls to the same function at once

B) Prevents a function from being called again before it finishes

C) Makes the contract upgradable

D) Disables all events

Answer: B

Explanation: The `nonReentrant` modifier blocks reentrant calls, protecting against a common vulnerability.

---

3. Why should you avoid using `tx.origin` for authentication?

A) It is more secure than `msg.sender`

B) It can be exploited by attackers to bypass security

C) It saves gas

D) It is required for ERC-721 contracts

Answer: B

Explanation: `tx.origin` can be exploited in phishing attacks; use `msg.sender` for authentication instead.

---

4. What is a good practice for validating user input in Solidity?

A) Trust all input

B) Use `require()` or `revert()` to check conditions

C) Ignore input validation

D) Only validate input after state changes

Answer: B

Explanation: Always validate user input with `require()` or `revert()` before making state changes.

Awesome! Neri's fortress is now secure against Hackana's most devious exploits.