Security & Auditing

```
contract BadRNG {
    address payable[] private s_players;

    function enterRaffle() external payable {
        require(msg.value >= 1000000000000000000);
        s_players.push(payable(msg.sender));
    }

    function pickWinner() external {
        uint256 randomWinnerIndex = uint256(
            keccak256(abi.encodePacked(block.difficulty, msg.sender))
        );
        address winner = s_players[randomWinnerIndex % s_players.length];
        (bool success, ) = winner.call{value: address(this).balance}("");
        require(success, "Transfer failed");
    }
}
```

```
contract MetamorphicContract is Initializable {
    address payable owner;

    function kill() external {
        require(msg.sender == owner);
        selfdestruct(owner);
    }
}
```