Uninitialized Storage Pointer

Description

Uninitialized local storage variables can point to unexpected storage locations in the contract, which can lead to intentional or unintentional vulnerabilities.

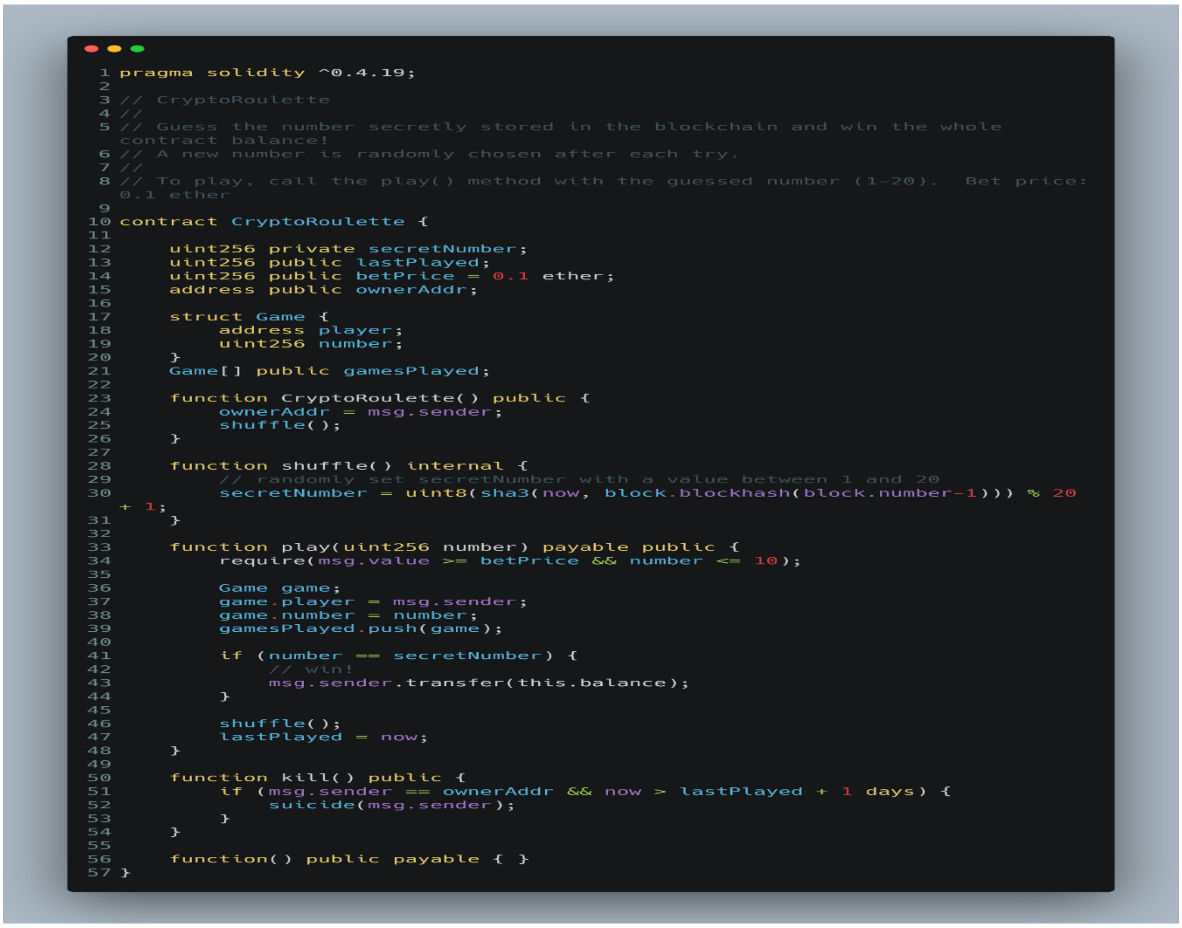
Remediation

Check if the contract requires a storage object as in many situations this is actually not the case. If a local variable is sufficient, mark the storage location of the variable explicitly with the memory attribute. If a storage variable is needed, then initialize it upon declaration and additionally specify the storage location storage.

Note: As of compiler version 0.5.0 and higher this issue has been systematically resolved as contracts with uninitialized storage pointers do no longer compile.

Example:

Code:



```
pragma solidity ^0.4.19;

// CryptoRoulette
//
// Guess the number secretly stored in the blockchain and win the whole contract balance!
// A new number is randomly chosen after each try.
//
// To play, call the play() method with the guessed number (1-20).  Bet price: 0.1 ether

contract CryptoRoulette {

    uint256 private secretNumber;
    uint256 public lastPlayed;
    uint256 public betPrice = 0.1 ether;
    address public ownerAddr;

    struct Game {
        address player;
        uint256 number;
    }
    Game[] public gamesPlayed;

    function CryptoRoulette() public {
        ownerAddr = msg.sender;
        shuffle();
    }

    function shuffle() internal {
        // randomly set secretNumber with a value between 1 and 20
        secretNumber = uint8(sha3(now, block.blockhash(block.number-1))) % 20 + 1;
    }

    function play(uint256 number) payable public {
        require(msg.value >= betPrice && number <= 10);

        Game game;
        game.player = msg.sender;
        game.number = number;
        gamesPlayed.push(game);

        if (number == secretNumber) {
            // win!
            msg.sender.transfer(this.balance);
        }

        shuffle();
        lastPlayed = now;
    }

    function kill() public {
        if (msg.sender == ownerAddr && now > lastPlayed + 1 days) {
            suicide(msg.sender);
        }
    }

    function() public payable { }
}
```

Explanation:

Local variables within functions default to storage or memory depending on their type. Uninitialized local storage variables can point to other unexpected storage variables in the contract, leading to intentional (i.e. the developer intentionally puts them there to attack later) or unintentional vulnerabilities.