Unencrypted Private Data On-Chain

Description

It is a common misconception that private type variables cannot be read. Even if your contract is not published, attackers can look at contract transactions to determine values stored in the state of the contract. For this reason, it's important that unencrypted private data is not stored in the contract code or state.

Remediation

Any private data should either be stored off-chain, or carefully encrypted.

Example:

Code:

```
pragma solidity ^0.5.0;

contract OddEven {
    struct Player {
        address addr;
        uint number;
    }

    Player[2] private players;
    uint count = 0;

    function play(uint number) public payable {
            require(msg.value == 1 ether, 'msg.value must be 1 eth');
            players[count] = Player(msg.sender, number);
            count++;
            if (count == 2) selectWinner();
    }

    function selectWinner() private {
            uint n = players[0].number + players[1].number;
            (bool success, ) =
  players[n%2].addr.call.value(address(this).balance)("");
            require(success, 'transfer failed');
            delete players;
            count = 0;
    }
}
```

Explanation:

If private variables are not encrypted, Block storage data would also access all of those variables.

Using the web3 API getStorageAt(), it is possible to get the block information that contained the private variable data.