Scope

The commit reviewed was a62bd1557615c97e5418d18b62ab83e2f3356eba. The review covered the entire repository at this specific commit but focused on the contracts directory. The review is a code review to identify potential vulnerabilities in the code. The reviewers did not investigate security practices or operational security and assumed that privileged accounts could be trusted. The reviewers did not evaluate the security of the code relative to a standard or specification. The review may not have identified all potential attack vectors or areas of vulnerability.The reviewer does not intent to perform re-review after fixes.

Findings

Explanation Findings are broken down into sections by their respective impact:

- Critical, High, Medium, Low impact
  - These are findings that range from attacks that may cause loss of funds, impact control/ownership of the contracts, or cause any unintended consequences/actions that are outside the scope of the requirements,
- Gas savings
  - Findings that can improve the gas efficiency of the contracts
- Informational
  - Findings including recommendations and best practices

High findings

1)
ID-0 LendingController.deposit(address,uint256,address,address) ignores return value by IERC20(_usdc).transferFrom(sender,address(this),_usdcAmount)
../src/LendingController.sol#L23-L27

Recommendation - replace code:

```
IERC20(_usdc).transferFrom(sender, address(this), _usdcAmount);
```

With:

```
(bool success,) =IERC20(_usdc).transferFrom(sender, address(this),
_usdcAmount);
require(success, "Transfer failed")
```

Medium findings

1)

unused-return

Impact: Medium Confidence: Medium

> ID-2 LendingController.deposit(address,uint256,address,address) ignores return value by IERC20(_usdc).approve(address(_aPool),_usdcAmount)

../src/LendingController.sol#L23-L27

> ID-3 LendingController.withdraw(address,uint256,address,address,address)ignores return value by ILendingPool(_aPool).withdraw(address(_usdc),_usdcAmount,sender)

../src/LendingController.sol#L33-L38

> ID-4 DrawController.fundVRFSubscription(uint96) ignores return value by link.transferAndCall(address(coordinator),amount,abi.encode(subscriptionId))

../src/DrawController.sol#L136-L138

> ID-5 LendingController.withdraw(address,uint256,address,address,address)ignores return value by IAToken(aToken).approve(address(_aPool),_usdcAmount)

../src/LendingController.sol#L33-L38

Recommendation - please refactor code to use return statement and and action in case of returned other flag than success

Low findings

1)

mpact: Low Confidence: Medium

ID-8 Reentrancy in DrawController.closeDraw(): External calls:

- requestId = coordinator.requestRandomWords(keyHash,subscriptionId,3,50000,2) Event emitted after the call(s):
- RandomnessRequested(requestId,currentDraw.drawId)

../src/DrawController.sol#L90-L97

ID-9 Reentrancy in DrawController.fulfillRandomWords(uint256,uint256[]): External calls:

- pool.mintTicket(currentUser,prize) Event emitted after the call(s):
- WinnerElected(drawId,currentUser,prize)

../src/DrawController.sol#L102-L122

ID-10 Reentrancy in PremiumPool.withdraw(uint256): External calls:

- lending.withdraw(msg.sender,_usdcAmount,address(usdc),address(aToken),address(aPool))
- ticket.burn(msg.sender,_usdcAmount) Event emitted after the call(s):
- Withdraw(msg.sender,_usdcAmount)

../src/PremiumPool.sol#L56-L68

ID-11 Reentrancy in PremiumPool.deposit(uint256): External calls:

- lending.deposit(msg.sender,_usdcAmount,address(usdc),address(aPool))
- (success) = address(this).call(abi.encodeWithSignature(mintTicket(address,uint256),msg.sender,_usdcAmount)) Event emitted after the call(s):
- Deposit(msg.sender,_usdcAmount)

../src/PremiumPool.sol#L38-L50

Recommendation:
Apply the check-effects-interactions pattern.
http://solidity.readthedocs.io/en/v0.4.21/security-considerations.html#re-entrancy

2)
Impact: Low Confidence: Medium

ID-7 Reentrancy in PremiumPool.withdraw(uint256): External calls:

- lending.withdraw(msg.sender,_usdcAmount,address(usdc),address(aToken),address(aPool))
- ticket.burn(msg.sender,_usdcAmount) State variables written after the call(s):
- userIndex[msg.sender] = 0
- delete users[userIndex[msg.sender]]

```
../src/PremiumPool.sol#L56-L68
```

Recommendation:
Apply the check-effects-interactions pattern.
http://solidity.readthedocs.io/en/v0.4.21/security-considerations.html#re-entrancy
3)
Impact: Low Confidence: High

ID-6
DrawController.constructor(address,address,uint64,bytes32).vrfCoordinatorshadows:

- VRFConsumerBaseV2.vrfCoordinator (state variable)

```
../src/DrawController.sol#L57
```

Recommendation
Rename the local variables that shadow another component.

Compilation warnings

Although not a direct security vulnerability, compilation warnings indicates missing opportunities to improve code clarity and readability which make development more error-prone. I would mark them as Medium findings:

```
warning[5667]: Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.
  --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:306:45:
   |
306 |   function requestSubscriptionOwnerTransfer(uint64 _subId, address _newOwner) external
pure override {
    |                                            ^^^^^^^^^^^^^



warning[5667]: Warning: Unused function parameter. Remove or comment out the variable
name to silence this warning.
   --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:306:60:
    |
306 |   function requestSubscriptionOwnerTransfer(uint64 _subId, address _newOwner) external
pure override {
    |                                                            ^^^^^^^^^^^^^^^^^



warning[5667]: Warning: Unused function parameter. Remove or comment out the variable
name to silence this warning.
   --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:310:44:
    |
310 |   function acceptSubscriptionOwnerTransfer(uint64 _subId) external pure override {
    |                                            ^^^^^^^^^^^^^



warning[5667]: Warning: Unused function parameter. Remove or comment out the variable
name to silence this warning.
   --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:314:33:
    |
314 |   function pendingRequestExists(uint64 subId) public view override returns (bool) {
    |                                 ^^^^^^^^^^^^



warning[2072]: Warning: Unused local variable.
  --> src/PremiumPool.sol:77:38:
   |
77 |         (, bool currentDrawIsOpen, , uint256 currentDrawEndTime, , , ) =
draw.draws(currentDrawId);
   |                                      ^^^^^^^^^^^^^^^^^^^^^^^^^^



warning[2018]: Warning: Function state mutability can be restricted to pure
```

```
  --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:261:3:
   |
261 |   function getConfig()
   |   ^ (Relevant source part starts here and spans across multiple lines).



warning[2018]: Warning: Function state mutability can be restricted to pure
  --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:274:3:
   |
274 |   function getFeeConfig()
   |   ^ (Relevant source part starts here and spans across multiple lines).



warning[2018]: Warning: Function state mutability can be restricted to pure
  --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:302:3:
   |
302 |   function getFallbackWeiPerUnitLink() external view returns (int256) {
   |   ^ (Relevant source part starts here and spans across multiple lines).



warning[2018]: Warning: Function state mutability can be restricted to pure
  --> lib/chainlink/contracts/src/v0.8/mocks/VRFCoordinatorV2Mock.sol:314:3:
   |
314 |   function pendingRequestExists(uint64 subId) public view override returns (bool) {
   |   ^ (Relevant source part starts here and spans across multiple lines).
```