# Security Audit Report

1. **Problem**: The for loop could create issue with the gas limits as it depends on the storage value and should be used carefully. During a malicious attack the for loop can grow exponentially and beyond the gas limit and can cause the contract to stop or revert.

**Solution**: In the loop we can add another condition where if gas left is below the limit then the loop should be ended there and then.

1. **Problem**: ‘this’ keyword in the payout function can cause unreliable results as if a low level call is made to the function the ‘this’ may refer to another function and then the balance will of the address calling the function.

**Solution**: msg.sender should be used in place of ‘this’ keyword or a self-defined variable should be used which will not be effected by the forced ether sent by selfdestruct.

1. **Problem**: The division of the balance to length could result in underflow situation.

**Solution**: Safemath library function for division should be used.

1. **Problem**: In the payout function a low level call is being made to payee call back and can provide the control to the caller and hereby making the contract available to be exploited.

**Solution**: Such low level calls should be made with great precision and care. Consideration should be made as to which address will be used to make this call.