| **S.No.** | **Issues** | **Issue type** | **Description** | **Resolution** |
| --- | --- | --- | --- | --- |
| **1** | Expired points are not Deducted | Medium severity issue | According to the contract’s logic, the supply of maximum points are capped (limited). Although, there is a check that the totalSupply+\_points (allocated or modified) must be less than or equal to maxSuppy but there is no removal of points after they are expired. However, if the expired points are not removed from the contract then the contract will neither be able to allocate any new points nor modify the existing ones. Hence, the contract would be inoperable. | The points are deducted whenever their expiry is checked    or when the points are redeemed. |
| **2** | Centralization Risk | Low severity issue | Using the “OnlyOwner” modifier means that only the owner of the contract will be able to call the functions regarding allocation of new points, modification of points, redeem points, etc.  It poses two risks, first one is in a scenario where the owner will lose their Private Key then in that case the contract will be frozen, will never be able to work again, and all the points could be lost because all the externally callable functions contain the “OnlyOwner” modifier. Secondly, the working can be very inefficient as only the user with the Owner’s private will be able to call the functions. | The business logic of the smart contract is inherently centralized. Only one person/entity is allowed to use the contract. However, to take care of the first risk of losing the private key, the modifier has been revised. Instead of a single owner of the contract, there can be multiple private keys that can be added and used. Lost keys or compromised accounts can be blacklisted/blocked. |
| **3** | Unlocked pragma | Informational issue | Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively. | The pragma of solidity has been locked to version 0.8.14 |
| **4** | Gas optimization | General recommendation | In conclusion, we would like to mention that some of the public functions that are never called from within the contract should be declared external in order to save gas. setOwner allocatePoints redeemUserPoints modifyCouponPoints | These functions have been set to be external. |