

REPORT 6052804315BAA80018C536E9

Created Wed Mar 17 2021 22:18:43 GMT+0000 (Coordinated Universal Time)  
Number of analyses 1  
User ikermacaya@gmail.com

## REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
<a href="#">d04f3f7e-6661-4e0a-a4eb-7c7a12f14cfb</a>	localhost/TFE/INDEX/CONTRACTS/SC_DAPP/DAPP.sol	6

Started	Wed Mar 17 2021 22:18:48 GMT+0000 (Coordinated Universal Time)
Finished	Wed Mar 17 2021 22:34:13 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Client Tool	Remythx
Main Source File	localhost/TFE/INDEX/CONTRACTS/SC_DAPP/DAPP.sol

DETECTED VULNERABILITIES



ISSUES

MEDIUM

Function could be marked as external.  
The function definition of "setDron" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

SWC-000

Source file  
localhost/TFE/INDEX/CONTRACTS/SC\_DAPP/DAPP.sol  
Locations

```
42 |
43 | // Dron set!
44 | function setDron(address company_, uint256 idDron_, uint256 altitudeMax_, uint256 altitudeMin_, uint256 costValue_, string memory pesticide_, string memory name_) public onlyOwner
45 | returns (bool) {
46 |     setDron(company_, idDron_, altitudeMax_, altitudeMin_, costValue_, pesticide_, name_);
47 |     return true;
48 | }
49 | function _setDron(address company_, uint256 idDron_, uint256 altitudeMax_, uint256 altitudeMin_, uint256 costValue_, string memory pesticide_, string memory name_) internal returns
50 | (bool) {
    drones[idDron_].company = company_;
    drones[idDron_].altitudeMax = altitudeMax_;
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "setLand" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

localhost/TFE/INDEX/CONTRACTS/SC\_DAPP/DAPP.sol

Locations

```
88 |
89 | // Land set!!
90 | function setLand(uint256 idLand_, uint256 altitudeMax_, uint256 altitudeMin_, address customer_, string memory pesticide_, string memory name_ public onlyOwner returns (bool) {
91 |     setLand(idLand_, altitudeMax_, altitudeMin_, customer_, pesticide_, name_);
92 |     return true;
93 | }
94 |
95 | function _setLand(uint256 idLand_, uint256 altitudeMax_, uint256 altitudeMin_, address customer_, string memory pesticide_, string memory name_) internal returns (bool) {
96 |     lands[idLand_]._altitudeMax = altitudeMax_;
97 |     lands[idLand_]._altitudeMin = altitudeMin_;
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "setPay" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

localhost/TFE/INDEX/CONTRACTS/SC\_DAPP/DAPP.sol

Locations

```
124 | mapping (bytes8 => Paid) private payments;
125 |
126 | function setPay(bytes8 purchase_, uint256 idDron_, uint256 idLand_, address idCostumer_, uint256 value_ public onlyOwner returns(bool) {
127 |     setPay(purchase_, idDron_, idLand_, idCostumer_, value_);
128 |     return true;
129 | }
130 |
131 | function _setPay(bytes8 purchase_, uint256 idDron_, uint256 idLand_, address idCostumer_, uint256 value_) internal onlyOwner returns(bool) {
132 |     //adiciona un require que valide el by_ no exista en payments
133 |     payments[purchase_]._idDron = idDron_;
```

## MEDIUM Function could be marked as external.

SWC-000

The function definition of "getDronAvailable" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

localhost/TFE/INDEX/CONTRACTS/SC\_DAPP/DAPP.sol

Locations

```
143 | }
144 |
145 | function getDronAvailable(uint256 max_, uint256 min_, string memory pesticide_) public view returns (uint256 IDDron_, uint256 altitudeMax_, uint256 altitudeMin_, uint256 costValue_) {
146 |     IDDron_ = LibraryDAPP.indexOf(drones.max_, min_, pesticide_);
147 |     altitudeMax_ = drones[IDDron_]_.altitudeMax;
148 |     altitudeMin_ = drones[IDDron_]_.altitudeMin;
149 |     costValue_ = drones[IDDron_]_.costValue;
150 | }
151 |
152 |
```

## LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.5.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

localhost/TFE/INDEX/CONTRACTS/SC\_DAPP/DAPP.sol

Locations

```
1 | //Sistema de fumigación parcelas con drones
2 |
3 | pragma solidity ^0.5.0;
4 |
5 |
6 | import "./IDAPP.sol";
```

## LOW

Unused state variable "payments".

SWC-131

The state variable "payments" is declared within the contract "DAPP" but its value does not seem to be used anywhere.

Source file

localhost/TFE/INDEX/CONTRACTS/SC\_DAPP/DAPP.sol

Locations

```
122 | //El contrato DAPP implementa un mapping (bytes32 => Paid) para almacenar la struct
123 | //El grupo de funciones a continuación describen la operación pago del servicio de fumigación
124 | mapping (bytes8 => Paid) private payments;
125 |
126 | function setPay(bytes8 purchase_, uint256 idDron_, uint256 idLand_, address idCostumer_, uint256 value_) public onlyOwner returns(bool) {
127 |     function setPay(bytes8 purchase_, uint256 idDron_, uint256 idLand_, address idCostumer_, uint256 value_) public onlyOwner returns(bool) {
128 |         _setPay(purchase_, idDron_, idLand_, idCostumer_, value_);
129 |     }
130 |     return true;
131 | }
```