AUDIT AND REPORT BY 0XDIT

AUDIT AUTHOR: 0X37SET

DISCLAIMER:

This audit is only to the Smart-Contract code at the specified address!
0xdit is a 3rd party auditing company that works on audits based on client requests & community request and as a professional auditing firm, we check on the contract for any vulnerabilities, backdoors, and/or scam scripts, investor lose and future Rugs.
Therefore:
We are not financial advisors nor do we partner with the contract owners
Operations and website administration are fully on the client's side

We do not have influence over client operations, which can lead to website changes, withdrawal function closes, etc. One always has the option to do this through the contract.
Any concerns about the project themselves need to be raised directly to the project owners and not through 0XDIT.
Investors are not in any way obliged, coerced, or influenced to invest in projects audited by 0XDIT.
We are not responsible for your funds or guarantee you profits.
We highly recommend that investors do their own research and gain crypto experience before investing

To report any scam, malpractices and irregularities, please send a message via Telegram to @kayie0x3

Smart Contract Audit by 0xDit:
https://bscscan.com/address/0xb8CDa6AE6D005d56205B29797ADD341c85e59608#code

PROJECT: https://www.yieldrobot.app/

SUMMARY

This smart contract has been reviewed by 0xdit and found one backdoor

function setCoupon(CouponSigData calldata coupon, Sig calldata sig) The message will be call once owner want because this smart contract is rewritable. You can see clearly this function is on line 1088.

re**quire(coupon.owner == msg.sender, "Not signature owner");** on line no 1092.

Once the owner set the couple amount, he can withdraw that amount. It's means if he set to withdraw all his contract funds. He/she can.

REVISION ON SMART CONTRACT

The smart contract can be upgrade by owner it's upgradable smart contract.

Fees: 3% deposit + 8% referral (Total 11%)

ROI: Daily 2% fixed.

Withdraw Period: 30 days
There is 2 wallets: Signer wallet, and dev Wallet

These wallets can be change anytime by the owner.

This contract is control of owner. Whatever he wants he can do. He doesn't control over the invest withdraw and these functions. Let me explain the code.

`function resetContract(address _devWallet) :` || Owner can change anytime dev wallet.

`function setSigner(address _signer) :` || Owner can change anytime signer wallet

`function deposit(uint256 _amount, address _referrer)` ||

This function has the referrer you can see it’s the referral address. In the UI there should be an option that people can by default put the referral address otherwise 80% traffic will be redirected to by default address which is given by the owner in the backend. If the function has two value for example (a,b) it’s mean both has to be enter then the function will work. In their smart contract (uint256 _amount, address _referrer) so it’s my request to the owner put the input field on the UI so people can know where their referral rewards are going.”

The fee will be deducting from you while depositing in the project which is 3% to dev and 8% to your refer person after all this deduction of 13%. 89% will goes to the smart contract

struct InvestorStruct{

address investor; // investor address

address referrer; // investor refer person address

uint256 totalLocked; // how much BUSD they have locked

uint256 startTime; it will store the time of your investment

uint256 lastCalculationDate; the deadline time of your withdrawal

uint256 claimableAmount; // the amount how much you are claiming

uint256 claimedAmount; // the amount how much you have claimed

uint256 referAmount; // how much is the referral fee on your investment

}

On the Deposit function there is no upgradable policy applied.

function claimAllReward() : || In this function you can claim your rewards. This function is completely scanned by me it has no exploit or any single pinny which is going to owner. In this function there is no upgradable policy applied.

function withdrawCapital(uint256 id) || In this function as you pass the current stake or whatever you have opened it will be passed here and it will see if your deadline is completed or not. If completed it will withdraw your funds. This function has no exploit at all or no upgradable policy applied on it.

function setCoupon(CouponSigData calldata coupon, Sig calldata sig) : || First of all let me explain to all my audit brothers that learn how the upgradable policy work on the functions. The upgradable policy is applied on this function it is a big warning that only owner can withdraw funds. Here is 100% exploit involved. Let me tell you that this function can be call by owner only. He can send the amount that how much he wants to take out of the smart contract.

On line no: 1092: require(coupon.owner == msg.sender, "Not signature owner"); You might not experience with the programing but you might see coupon.owner the owner can set this message and put the amount how much he want to withdraw by coupon. If he set his address or

someone address and allow him how much they want to withdraw so they can.

Look at here carefully:

```
if(coupon.payAmount > 0) {

    totalInvested = totalInvested.add(coupon.payAmount);


    IERC20(BUSDContract).safeTransferFrom(msg.sender, address(this), coupon.payAmount);
```

Owner can set the coupon and withdraw that coupon on line no 1115.

SCORE

TEAM: 0/10: - (The team has fake KYC as I have confirmed about it)

Smart Contract: 0/10: - (Upgradable and Exploit in Coupon)

Exploit 0/10: - (There is no exploit and backdoor in the smart contract)

Funds Generating: 2/10 (It’s depended on investors’ money. Straightforward Ponzi)

RISK: 0/10 - (Exploit and Fake KYC)

FEES: 7/10 (The Fees are good 3% Deposit fee and 8% referral fee)

Total Score: 9/60