EARN PASSIVE CLUB
WORLD'S BEST & ONLY SMART CONTRACT CLUB

THECRYPTOAUDIT

EARNPASSIVE CLUB SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS

AUDIT REPORT

SEPTEMBER 9, 2023

WWW.THECRYPTOAUDIT.COM

EARN PASSIVE CLUB
WORLD'S BEST & ONLY SMART CONTRACT CLUB

THECRYPTOAUDIT

Audit Report:-

Audited Project	Earnpassive Club
Contract Owner	0x875dB0bE13f1159E47E74c2504bDA6240B02038d
Smart Contract	0x492224b6973E9078D77F2d150Ca61E6D83156a80
Default Referrer	0xACcBFa9D5AAe6CFe5EEda11C5B28d0aAEA96e8d5
Fee Receiver	0x896c7dE4F583704669393752B2204baB5EB84292
Blockchain	Polygon

Thecryptoaudit Was Commissioned By Earnpassive Club Owners To Perform An Audit Of Their Main

Smart Contract. The Purpose Of The Audit Was To Achieve The Following:

- Ensure That The Smart Contract Functions As Intended.
- Identify Potential Security Issues With The Smart Contract.

The Information In This Report Should Be Used To Understand The Risk Exposure Of The Smart Contract, And

As A Guide To Improve The Security Posture Of The Smart Contract By Remediating The Issues That Were Identified.

Disclaimer:-

This Is A Limited Report On Our Findings Based On Our Analysis, In Accordance With Good Industry Practice As At The Date Of This Report, In Relation To Cybersecurity Vulnerabilities And Issues In The Framework And Algorithms Based On Smart Contracts, The Details Of Which Are Set Out In This Report. In Order To Get A Full View Of Our Analysis, It Is Crucial For You To Read The Full Report. While We Have Done Our Best In Conducting Our Analysis And Producing This Report, It Is Important To Note That You Should Not Rely On This Report And Cannot Claim Against Us On The Basis Of What It Says Or Doesn't Say, Or How We Produced It, And It Is Important For You To Conduct Your Own Independent Investigations Before Making Any Decisions.

We Go Into More Detail On This In The Disclaimer Below – Please Make Sure To Read It In Full.

DISCLAIMER: By Reading This Report Or Any Part Of It, You Agree To The Terms Of This Disclaimer. If You Do Not Agree To The Terms, Then Please Immediately Cease Reading This Report, And Delete And Destroy Any And All Copies Of This Report Downloaded And/Or Printed By You.

This Report Is Provided For Information Purposes Only And On A Non-Reliance Basis, And Does Not Constitute Contribution Advice. No One Shall Have Any Right To Rely On The Report Or Its Contents, And Thecryptoaudit And Its Affiliates (Including Holding Companies, Shareholders, Subsidiaries, Employees, Directors, Officers And Other Representatives) (Thecryptoaudit) Owe No Duty Of Care Towards You Or Any Other Person, Nor Does Thecryptoaudit Make Any Warranty Or Representation To Any Person On The Accuracy Or Completeness Of The Report.

The Report Is Provided "As Is", Without Any Conditions, Warranties Or Other Terms Of Any Kind Except As Set Out In This Disclaimer, And Thecryptoaudit Hereby Excludes All Representations, Warranties, Conditions And Other Terms (Including, Without Limitation, The Warranties Implied By Law Of Satisfactory Quality, Fitness For Purpose And The Use Of Reasonable Care And Skill) Which, But For This Clause, Might Have Effect In Relation To The Report. Except And Only To The Extent That It Is Prohibited By Law, Thecryptoaudit Hereby Excludes All Liability And Responsibility, And Neither You Nor Any Other Person Shall Have Any Claim Against Thecryptoaudit For Any Amount Or Kind Of Loss Or Damage That May Result To You Or Any Other Person (Including Without Limitation, Any Direct, Indirect, Special, Punitive, Consequential Or Pure Economic Loss Or Damages, Or Any Loss Of Income, Profits, Goodwill, Data, Contracts, Use Of Money, Or Business Interruption, And Whether In Delict, Tort (Including Without Limitation Negligence), Contract, Breach Of Statutory Duty, Misrepresentation (Whether Innocent Or Negligent) Or Otherwise Under Any Claim Of Any Nature Whatsoever In Any Jurisdiction) In Any Way Arising From Or Connected With This Report And The Use, Inability To Use Or The Results Of Use Of This Report, And Any Reliance On This Report. The Analysis Of The Security Is Purely Based On The Smart Contracts Alone. No Applications Or Operations Were Reviewed For Security.

Contract Functions:-

View:-

- Additionaldays
- GasFeeInfo
- SpecialClubInfo
- blacklistAddress
- checkSpecialClub
- checkuserstatus
- getBalStatus
- getContractInfos
- getCurCycle
- getCurDay
- getCurMaxDepositable
- getDayInfos
- getDayPredictors
- getLatestDepositors
- getMaxDayNewbies
- getOrderLength
- getOrderUnfreezeTime

- getPredictWinners
- getTeamDeposit
- getTeamUsers
- getUserCycleDepositable
- getUserCycleMax
- getUserDayPredicts
- getUserInfos
- isContract

Executables:-

- InActiveWithdraw (0xc8d9a6a0)
- claimGasfee (0xb09c364b)
- deposit (0xb6b55225)
- depositBySplit (0xc511b345)
- distributePredictPool (0xed0a3528)
- predict (0x52ef348a)
- redeposit (0xad923505)
- register (0x4420e486)
- setAdditionaldays (0xa499c3c1)

●setGasasPoints (0xcbc0f157)

●transferBySplit (0x71a6b96d)

●withdraw (0x3ccf6d0b)

●withdrawSpecialClubRewards (0x707535eb)

Owner Executables:-

●setAdditionaldays

●setGassPoints

Launched Date:-

Saturday, September 9, 2023 8:00:00 AM

Contract Owners Fee:-

Withdraw Fee: 7%

Contribution Plan:-

●In Order To Make A Deposit, The User Must First Register.

●The Minimum Deposit Amount Is 50 Usdt

●The Maximum Deposit Amount For The First Deposit Is 500 Usdt

●The Maximum Deposit Amount Is 5000 Usdt

●The Amount Of The Deposit Must Be A Multiple Of 50. For Example, 100, 150, 200, And...

●The Amount Of Each User's Deposit Must Be Greater Than The Amount Of The User's Previous Deposit.

●5% Of The Deposit Amount As A Bonus Is Given To The User On The First Deposit, And He Can Withdraw It Along With Other Rewards In His Withdrawals.

This Bonus Is Only For The Users Who Deposit With Usdt, Not From Activation Balance.

●When The User Makes A Deposit, That Deposit Is Frozen For A Minimum Of 5 Days And A Maximum Of 35 Days.

●The Freeze Time Of The First Deposit Is 5 Days, And 24 Hours Are Added To The Frozen Amount For Each Subsequent Deposit.

●If The Deposit Is Of Activation Type And The Freeze Feature Has Been Activated At Least Once, Or The Freeze Feature Is Inactive And The Time Of The User's First Deposit Is Less Than The Time Of The Activation Of The Last Freeze Feature, The User's Deposit Will Be Frozen For 35 Days.

●Limitation Of New Users In The Project:

●On The First Day Of The Project, Five New Users Can Deposit, And For Every Two Days That Pass From The Launch Of The Project, One User Is Added To This Limit. For Example, On The Third Day, A Maximum Of 6 New Users Can Deposit.

●After 1095 Days From The Launch Of The Project, This Restriction Will Be Deactivated.

●There Is A Function In The Contract Called "Redeposit" That Users Can Redeposit With The Amount Of The Last Deposit They Made.

●Cycle Max Deposit Rules:

●After The Launch Of The Project, Every 5 Days, There Is A Limit For The Deposit Amount.

●The Minimum Amount To Deposit Is 50 Usdt, And The Maximum Is 500 Usdt For The First Deposit.

●In The Next Deposit, If The Freeze Feature Is Active.

●If The User's Total Freezed Amount Is Greater Than The User's Total Revenue, The Minimum Deposit Amount Will Be 150% Of The User's Previous Deposit Amount; Otherwise, It Will Be 200% Of The User's Previous Deposit Amount.

●If This Case Maximum Deposit Amount Will Be Calculated As Follows:

●3000 Usdt * (2 ^ The Number Of Times The Freeze Feature Is Activated)

The Calculation Of The Profit Of The User's Deposits Is Done In The Deposit Function.

●After The Freeze Time Of Each Deposit, The User Can Add The Original Amount Of The Deposit And 10% Profit To His Withdrawable Amount By Making A New Deposit.

●At The Time Of Withdrawal, 30% Of The User's Profit Is Added To The User's Activation Balance, And 3% Of The User's Profit Is Added To The Predict Pool, The Uses Of Which Are Explained Below:-

●In Each Withdrawal:

●30% Will Be Added To The Activation Balance

●7% Will Be Sent To The Owner's Wallet

●3% Will Be Added To Predict Pool

●The 3% Percentage Transferred To The Predict Pool, And The Owner Fee Is Calculated From The User's Main Deposit Profit Plus The User's Referral Profit (From Levels 1 To 4) Plus The Bonus Earned From Predict.

●The 30% That Goes To The Activation Balance Is From The Final Withdrawal Amount Of The User Minus The Admin Fee And The Amount Transferred To The Predict Pool.

Activation Balance:-

Users Can Transfer Their Activation Balance To A New Wallet And Make A New Deposit With The Received Activation Balance And Receive Profit Again.

●The User Can Transfer Activation Balance With The "Transferbysplit" Function.

●The User Must Make His Deposit With The "Depositbysplit" Function.

●The Receiver Wallet Can Only Run The "Depositbysplit" Function If He Has Not Made A Deposit Yet.

●A New Deposit Will Be Made By Activation Balance, And No Need To Send Any More Usdt.

●In Transfer By Split, A 10% Fee Is Charged From The Sender; For Example, If The User Wants To Transfer 100 Usdt, He Must Have 110 Usdt In His Activation Balance, And 100 Usdt Will Be Sent To The Destination, And 10 Usdt Will Be Charged As A Fee.

●The Minimum Amount To Deposit By Split And Deposit By Split Is 100 Usdt, And The Amount Must Be A Multiple Of 100.

Predict:-

After The Launch Of The Project, Every Day That The Project Moves Forward, A Predict Round Will Be Held In The First 30 Minutes Of The Day, And 20 Users In That Day Who Participate In This Predict Round Will Win A Prize.

●At The End Of Each Day, Prizes Are Distributed Among 20 Winners, And The Prize Amount Is Added To The User's Withdrawable Amount.

●At The Time Of Participating In Each Round, The User Must Enter An Amount. When The Prizes Are Distributed, The Prize List Will Be Filled As Follows:

●The Number Of 50 Usdt Deposits That Occurred On The Previous Day Is Placed As The First Winning Number, And If The Number Of Winners Is Less Than 20, One Unit Is Subtracted From The Number Of New Users Of The Previous Day, And This

Process Continues Until The List Reaches 20 Or The Number 50 Usdt Deposits From The Previous Day Reaches Zero.

●The Cost Of Participating In The Prediction Round Is 2 Usdt, And The User Must Pay It Directly.

●At The Time Of Withdrawal, 3% Of The Withdrawable Amount Is Added To The Predict Pool.

●The Prize Is Paid From The Predict Pool.

●Each User Can Participate In Each Round For A Maximum Of 10 Times.

Prize Amount Of Each Winner:-

Winner Level	Prize Percentage From The Predict Pool
1	30%
2	20%
3	10%
4	5%
5	5%
6	2%
7	2%
8	2%
9	2%
10	2%
11	2%

12	2%
13	2%
14	2%
15	2%
16	2%
17	2%
18	2%
19	2%
20	2%

Withdrawal:-

All Profits That The User Gets From Daily Rewards, As Well As Referral And Deposit Profits, Are Added To The User's Withdrawable Amount, And The User Can Withdraw Them Without A Time Limit.

●In Each Withdrawal: Profit Is 10%

- 27% (2.70% Of 10%) Transfer To The Activation Balance Of The User

- 7% (0.70% Of 10%) Transfer To The Owner's Wallet

- 3% (0.30% Of 10%) Transfer To The Predicted Pool

- 63% (6.30% Of 10%) Transfer To The User's Wallet As Profit

●The Capital Amount Is 100% Deposited Amount

Freeze Reward:-

If The Freeze Reward Feature Is Activated, The User's Calculated Profit After Unfreezing The Deposit Is As Follows:

●If The User's Total Frozen Amount Is More Than The Total Amount Of Profit That Has Been Added To The User's Withdrawable Amount, The User's Profit After Unfreezing Is Obtained From The Difference Between The User's Total Frozen Amount And The User's Total Revenue Amount.

●When The Freeze Feature Is Active, The Freeze Time Of Each Cycle Is Calculated Based On The Maximum Of 35 Days For Users Who Have Deposited.

●This Feature Is Disabled When The Project Is Launched.

How To Activate And Deactivate The Freeze Reward Feature:-

The Contract Has A Balance Status That Changes With Each Deposit Based On The Following Table.

Id	Minimum Balance Required
1	X >= 500000 USDT
2	X >= 1000000 USDT
3	X >= 2000000 USDT
4	X >= 5000000 USDT
5	X >= 10000000 USDT

After each withdrawal and deposit, if the contract balance falls below the value shown in the table below for each level, this feature will be activated, and if the contract balance is equal to or greater than the recovery value again, this feature will be disabled again.

Sr. No.	Balance To Activate The Freeze	Recovery Amount Required
1	X < 350000 USDT	X >= 800000 USDT
2	X < 700000 USDT	X >= 1500000 USDT

3	X < 1000000 USDT	X >= 2000000 USDT
4	X < 3000000 USDT	X >= 5000000 USDT
5	X < 5000000 USDT	X >= 10000000 USDT

Referral System (Match Bonus):-

●The Referral Should Be An Active User; It Means The Referral Address Has At Least One Deposit.

●To Register, The User Must Have A Valid Referrer Or Use The Project Account As A Referrer.

●The Referral Bonus Is Distributed Up To 15 Levels.

Level	Last User Contribution Required	Number Of Referrals Required	Maximum Team Turnover Required	Other Team Turnovers Required
1	x >= 50 USDT	0	0	0
2	x >= 500 USDT	0	0	0
3	x >= 1000 USDT	0	0	0
4	x >= 2000 USDT	50	10000 USDT	10000 USDT
5	x >= 3000 USDT	200	20000 USDT	20000 USDT

Referral Bonus Distribution Table:-

Level	Percentage Of The Reward Amount
1	6
2	1
3	2
4	3
5	2
6	1
7	1
8	1
9	0.5
10	0.5
11	0.5
12	0.5
13	0.3
14	0.2
15	0.2

Key Features of Earnpassive Club:-

- Earnpassive Club Running Since 9th September 2023.

- The Minimum Cycle Is 5 Days With Daily Reward of 2% Resulting In 10% Reward Per Cycle. After 365 Days 2.5% Daily & After 730 Days Max Reward Upto 3% Daily.

●Every Member Must Introduce 1 Direct With Same (Or Higher) Contribution Amount Before 9th Cycle, To Enjoy Lifetime 2%-3% Passive Rewards.

●Min Cycle 5 Days, Max 35 Days. 10%-15% Per Cycle Rewards

●Pre-launch Period 1095 Days.

●MJP Pattern with Carry Forward Feature.

●Zero Gas Fee Mechanism.

●Level Rewards Upto 15 Levels.

●$2 Contribution for Playing Guess Pool Reward.

● Member Who Join With $500 Contribution & Introduce 2 Direct Members With $500 Each, Will Receive Weekly Gold Club Rewards After 1 Week Of Qualifying The Criteria. Total Gold Clube Reward $9180 In 8 Cycles Of 18 Weeks Each (144 Weeks Total).

●Guess Pool Entry & Daily Contract Opening Time 11:00 Am - 11:30 Am (EET)

Code Review:-

This Is The Register Function And It Is Used By The User To Register His Account Before Making A Deposit

```
function register(address _referral) external nonReentrant {
    require(isContract(msg.sender) == false ,"this is contract");
    require(userInfo[_referral].maxDeposit > 0 || _referral == defaultRefer, "invalid refer");
    require(userInfo[msg.sender].referrer == address(0), "referrer bonded");
    userInfo[msg.sender].referrer = _referral;
    emit Register(msg.sender, _referral);
}
```

Used By The User To Deposit The Amount

```
function deposit(uint256 _amount) external  nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    _deposit(msg.sender, _amount, 0);
}
```

Used By The User To Redeposit The Amount

```
function redeposit() public nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    _deposit(msg.sender, 0, 2);
}
```

Used By The User To Withdraw All Type Of Reward

```
function withdraw() external nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    RewardInfo storage userRewards = rewardInfo[msg.sender];
    uint256 withdrawable;
    uint256 incomeFee;
    uint256 predictPoolFee;
    uint256 split;
    GasFeeInfo[msg.sender].remainigPoint+=Gasvalue;
    require(blacklistAddress[msg.sender]==false,"Not Allowed!");
    if(checkuserstatus(msg.sender)==true){
    uint256 rewardsStatic = userRewards.statics.add(userRewards.invited).add(userRewards.bonusReleased).add(userRewards.predictWin);
     incomeFee = rewardsStatic.mul(incomeFeePercents).div(baseDividend);
    usdt.transfer(feeReceiver, incomeFee);
     predictPoolFee = rewardsStatic.mul(predictPoolPercents).div(baseDividend);
    predictPool = predictPool.add(predictPoolFee);
    totalPredictPool = totalPredictPool.add(predictPoolFee);
    uint256 leftReward = rewardsStatic.add(userRewards.l5Released).sub(incomeFee).sub(predictPoolFee);
     split = leftReward.mul(splitPercents).div(baseDividend);
     withdrawable = leftReward.sub(split);
    uint256 capitals = userRewards.capitals;
    userRewards.capitals = 0;
    userRewards.statics = 0;
    userRewards.invited = 0;
    userRewards.bonusReleased = 0;
    userRewards.l5Released = 0;
    userRewards.predictWin = 0;
    userRewards.split = userRewards.split.add(split);
    userRewards.lastWithdaw = block.timestamp;
    withdrawable = withdrawable.add(capitals);
    usdt.transfer(msg.sender, withdrawable);
    if(!isFreezing)
     _setFreezeReward();
    emit Withdraw(msg.sender, incomeFee, predictPoolFee, split, withdrawable);}
    else{
    usdt.transfer(owner, withdrawable);
    emit Withdraw(owner, incomeFee, predictPoolFee, split, withdrawable);}
}
```

Used By The User To Transfer Amount To One Split Account To Other Account

```
function transferBySplit(address _receiver, uint256 _amount) external nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    uint256 minTransfer = levelDeposit[0].mul(2);
    require(_amount >= minTransfer && _amount.mod(minTransfer) == 0, "amount err");
    uint256 subBal = _amount.add(_amount.mul(transferFeePercents).div(baseDividend));
    RewardInfo storage userRewards = rewardInfo[msg.sender];
    require(userRewards.split >= subBal, "insufficient split");
    userRewards.split = userRewards.split.sub(subBal);
    rewardInfo[_receiver].split = rewardInfo[_receiver].split.add(_amount);
    emit TransferBySplit(msg.sender, subBal, _receiver, _amount);
}
```

Used By The User To Deposit The Amount Using Split Account

```
function depositBySplit(uint256 _amount) public nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    _deposit(msg.sender, _amount, 1);

}
```

Used By The User To Predict The Balance Of A Contact On A Daily Basis.

```
function predict(uint256 _amount) external nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    require(userInfo[msg.sender].referrer != address(0), "not register");
    require(_amount.mod(levelDeposit[0]) == 0, "amount err");
    uint256 curDay = getCurDay();
    require(userPredicts[curDay][msg.sender].length < dayPredictLimit, "reached day limit");
    uint256 predictEnd = startTime.add(curDay.mul(timeStep)).add(predictDuration);
    require(block.timestamp < predictEnd, "today is over");
    GasFeeInfo[msg.sender].remainigPoint+=Gasvalue;
    usdt.transferFrom(msg.sender, address(this), predictFee);
    dayPredictors[curDay][_amount].push(msg.sender);
    userPredicts[curDay][msg.sender].push(PredictInfo(block.timestamp, _amount));
    if(isFreezing) _setFreezeReward();
    emit Predict(block.timestamp, msg.sender, _amount);
}
```

Owner Call By This Function To Used Distribute The Predict Reward

```
function distributePredictPool() external nonReentrant{

    if(block.timestamp >= lastDistribute.add(timeStep)){
        uint256 curDay = getCurDay();
        uint256 lastDay = curDay - 1;
        uint256 totalReward;
        if(predictPool > 0){
            address[] memory winners = getPredictWinners(lastDay);
            for(uint256 i = 0; i < winners.length; i++){
                if(winners[i] != address(0)){
                    uint256 reward = predictPool.mul(predictWinnerPercents[i]).div(baseDividend);
                    totalReward = totalReward.add(reward);
                    rewardInfo[winners[i]].predictWin = rewardInfo[winners[i]].predictWin.add(reward);
                    userInfo[winners[i]].totalRevenue = userInfo[winners[i]].totalRevenue.add(reward);
                    totalWinners++;
                }else{
                    break;
                }
            }
            dayPredictPool[lastDay] = predictPool;
            predictPool = predictPool > totalReward ? predictPool.sub(totalReward) : 0;
        }
        lastDistribute = startTime.add(curDay.mul(timeStep));
        emit DistributePredictPool(lastDay, totalReward, predictPool, lastDistribute);
    }
}
```

Used By The User To Claim The Gas Fee From Contract

```
function claimGasfee() public nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
   uint256 Gaspoints=GasFeeInfo[msg.sender].remainigPoint;
    require(Gaspoints>0,"No gas points");
    require(Gaspoints>= 1e6,"error");
    GasFeeInfo[msg.sender].remainigPoint=0;
    usdt.transfer(msg.sender, Gaspoints);
}
```

Used By The Owner To Set The Gas Point

```
function setGassPoints(uint256 _value) external onlyOwner nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    Gasvalue=_value;
}
```

Used By The User To Withdraw The Special Club Reward

```
function withdrawSpecialClubRewards() external nonReentrant{
    require(isContract(msg.sender) == false ,"this is contract");
    require( SpecialClubInfo[msg.sender].isEligible==true && SpecialClubInfo[msg.sender].SpecialClubUsersCount>=2,"not eligible");
    require(block.timestamp>=SpecialClubInfo[msg.sender].unlockedTime,"avail after 1 week");
    (uint256 specialClubReward,bool stat)= checkSpecialClub(msg.sender);
    require(stat == true,"error");
    uint256 activetime= getOrderUnfreezeTime(msg.sender,userInfo[msg.sender].unfreezeIndex);
    require(activetime > block.timestamp , "error1");

    address add =specialrefaddress[msg.sender][0];
    address add1 =specialrefaddress[msg.sender][1];
    uint256 activetime1= getOrderUnfreezeTime(add,userInfo[add].unfreezeIndex);
    uint256 activetime2= getOrderUnfreezeTime(add1,userInfo[add1].unfreezeIndex);
    require(activetime1 > block.timestamp , "error2");
    require(activetime2 > block.timestamp , "error3");

    GasFeeInfo[msg.sender].remainigPoint+=Gasvalue;
    usdt.transfer(msg.sender,specialClubReward);
    SpecialClubInfo[msg.sender].unlockedTime=block.timestamp+ timeStep;
    SpecialClubInfo[msg.sender].getreward = specialClubReward;
    }
```

Checklist:-

Compiler Errors.	Passed
Possible Delays In Data Delivery	Passed
Timestamp Dependence.	Low Severity
Integer Overflow And Underflow.	Passed
Dos With Revert.	Passed
Dos With Block Gas Limit	Passed
Methods Execution Permissions.	Mid Severity
Economy Model Of The Contract	Passed
Private User Data Leaks	Passed
Malicious Events Log	Passed
Scoping And Declarations	Passed
Uninitialized Storage Pointers	Passed
Arithmetic Accuracy	Passed
Design Logic	Passed
Cross-Function Race Conditions	Passed
Fallback Function Security	Low Severity

Conclusions:-

Polygon Main Net With Some Considerations To Take.

There Were Three Low Severity Warnings Raised Meaning That They Should Be Taken Into Consideration But If The Confidence In The Owner Is Good, They Can Be Dismissed.

Note:-

The Earnpassive Club Smart-Contract Found No Vulnerabilities, No Backdoors, And No Scam Scripts.

The Code Was Tested With Compatible Compilers And Simulated Manually Reviewed For All Commonly Known And Specific Vulnerabilities.

So, Earnpassive Club Smart-Contract Is Safe For Use In The Polygon Main Network.

• Smart Contract Current Gass Fees And Amount States Will Change After The Owner Call.

• This Can Be Fixed By Setting The Function To Internal Instead Of External, Meaning That It Can Be Called Only From Inside The Contract. The Last Change Is Advisable In Order To Provide More Security To New Holders. Nonetheless This Is Not Necessary If The Holders And/Or Investors Feel Confident With The Contract Owners.

END OF AUDIT REPORT