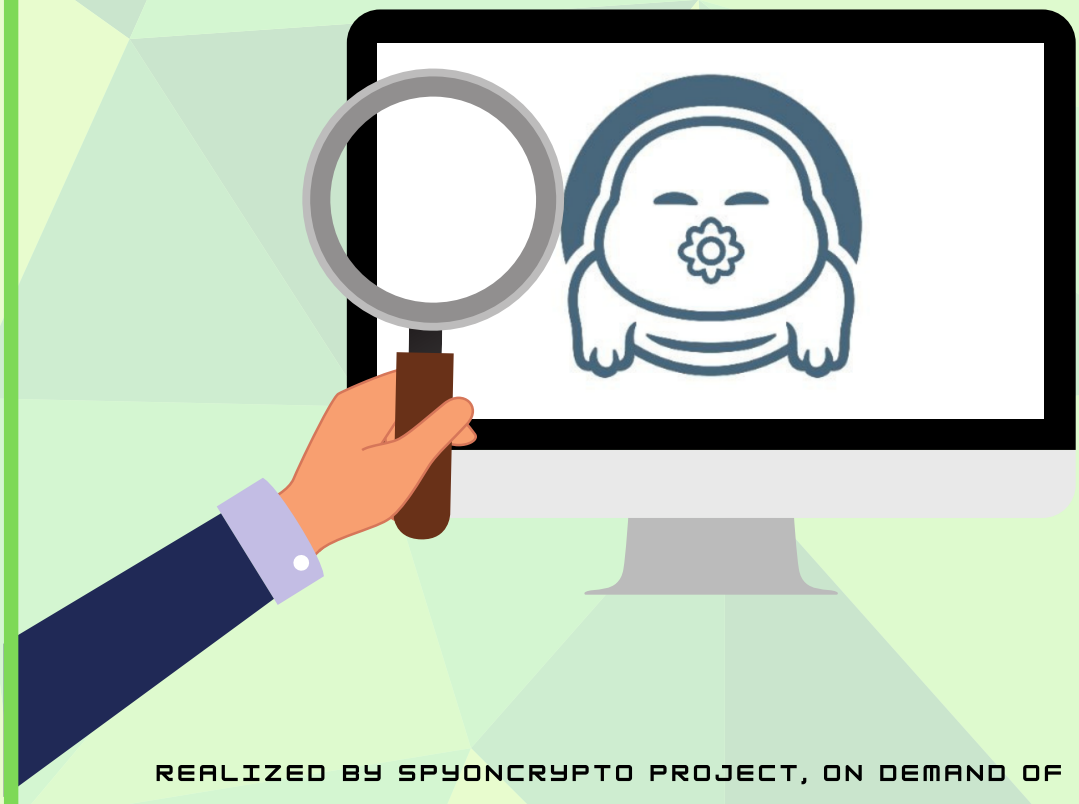
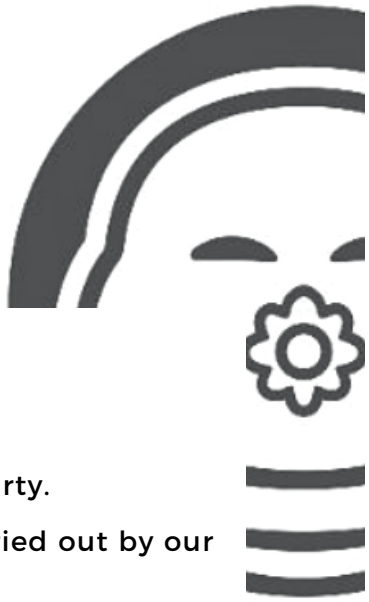
SPY ON CRYPTO

AUDIT TRDG



REALIZED BY SPYONCRYPTO PROJECT, ON DEMAND OF TRDG TEAM

DISCLAIMER



This file is an audit carried out at the request of the interested party.

This report is based on a multitude of analyses and research carried out by our team according to a predefined scheme.

The various steps set out in this file will make it possible to display any vulnerabilities relating to the cybersecurity of the project studied.

These searches are based on the information available to us through the smart contract, but also through information provided by the project developers.

In order to have a better overview of the possible vulnerabilities of this project, the complete reading of this file is recommended.

However, even if this report is available to you, it is only an additional element that can help you in your investigations.

Although a great deal of background work has been done in our investigations, we may have missed some elements, so further research on your part is necessary and advisable.

The conditions mentioned above in the disclaimer are not optional, so if you are not satisfied with them, we strongly urge you to stop reading and analyzing this file and to destroy any copies you have downloaded and/or printed.

These analyses and conclusions are not intended as investment advice. SpyonCrypto is not responsible for any loss of capital, which you are the only owner of.

This report is provided to you as, and without any conditions guaranteed.

SpyonCrypto disclaims any and all liability to the law for any claim or demand by you or any other person for damages.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security.

No product code has been reviewed.

SUMMARY

PRESENTATION

$TRDG has already established itself as a key player in the BINANCE SMART CHAIN.

In just a few months of existence, its name has become one of the most talked about topics in a multitude of telegram groups and twitter accounts.

Today, it has over 100k holders, and many fans around the world.

The goal of this project is to create an eco-system around the token in order to revolutionize the blockchain world.

- Holder structure shows burn address in 1st position, 2 external wallet and Pancakeswap in 4th position
- Ownership has been renounced
- No backdoors or DoS possible from the owner perspective
- Fixed tx fee of 5%
- 64% of total supply have been burned
- 98.8% liquidity is permanently burned

CONTRACT DETAILS

CONTRACT NAME
TARDIGRADES.FINANCE

SUBMITTED FOR VERIFICATION AT BSCSCAN
2021-05-18

CONTRACT ADDRESS
0X92A42DB88ED0F02C71D439E55962CA7CAB0168B5

TOTAL SUPPLY
100,000,000,000,000,000

TOKEN TICKER
TRDG

DECIMALS
9

TOKEN HOLDERS
101589

TRANSACTIONS COUNT
279258

TOP 100 HOLDERS DOMINANCE
78.76 %

CONTRACT DEPLOYER ADDRESS
0X1E15619B23B402B7950FDF0B9C94607D8B4DFD5A

CONTRACT'S CURRENT OXNER ADDRESS
0X0000000000000000000000000000000000000000

CURRENT LIQUIDITY FEE
5%

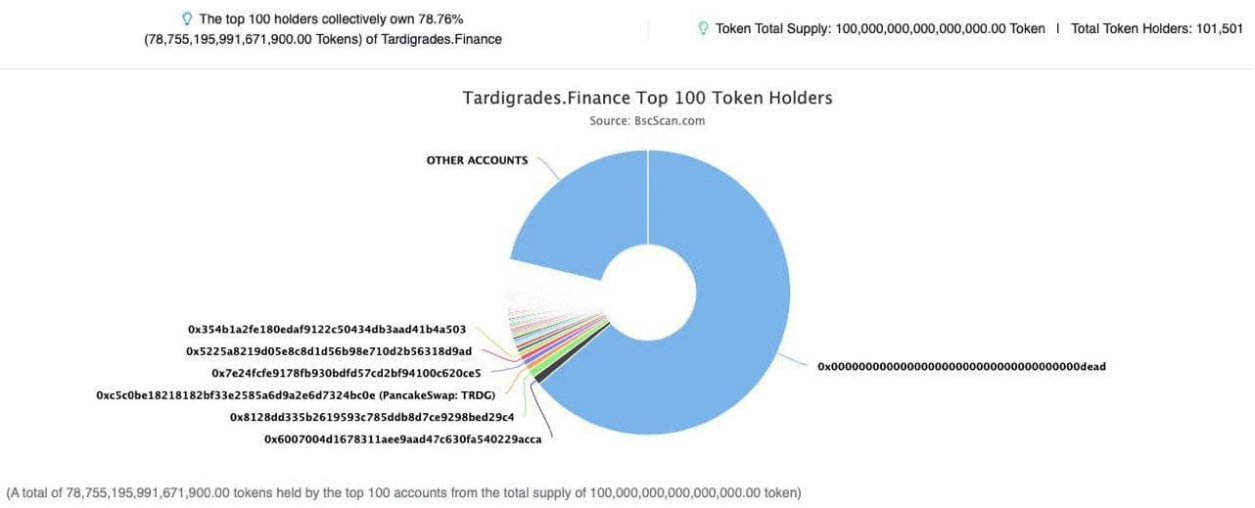
CURRENT TAX FEE
5%

TOTAL FEES
295748520607804213206472 25

DEPLOYED AT TRANSACTION
0XF1403E77B92CCB1E0A80224BDC02663D8597B49EB1EF8320E62E4A751E D58541

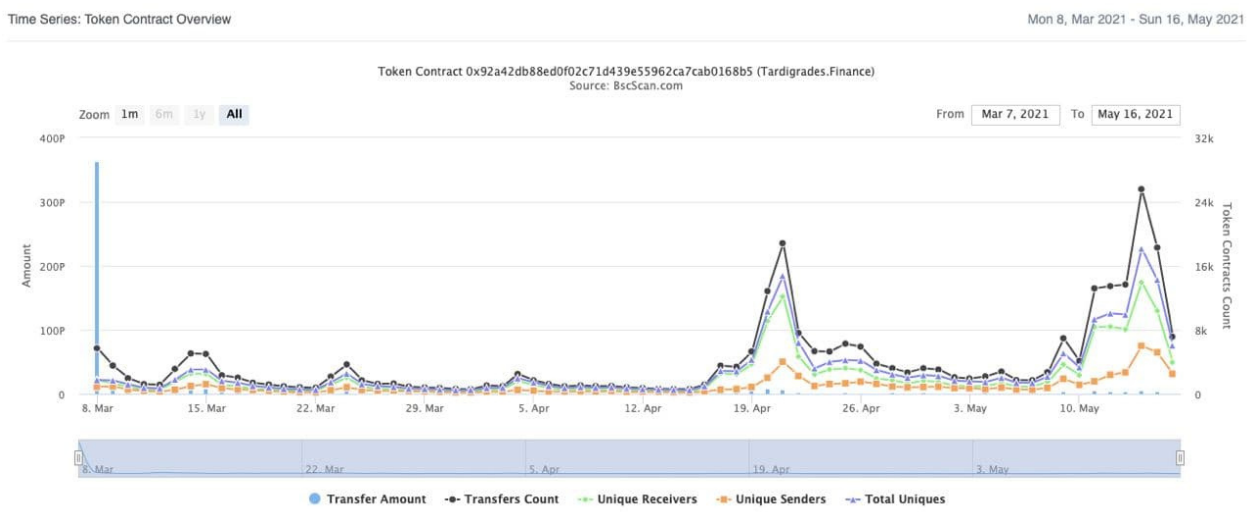
GRAPHIC ANALYSIS

"TRDG" Token distribution



(A total of 78,755,195,991,671,900.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

"TRDG" contract interaction details



DETECTED VULNERABILITIES

0

20

1

SECURITY ISSUES

MEDIUM

Function could be marked as external.
The function definition of "owner" 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.

```
386       */
387      function owner() public view returns (address) {
388          return _owner;
389      }
```

```
405       */
406      function renounceOwnership() public virtual onlyOwner {
407          emit OwnershipTransferred(_owner, address(0));
408          _owner = address(0);
409      }
```

```
414       */
415      function transferOwnership(address newOwner) public virtual onlyOwner {
416          require(newOwner != address(0), "Ownable: new owner is the zero address");
417          emit OwnershipTransferred(_owner, newOwner);
418          _owner = newOwner;
419      }
```

```
448
449      function name() public view returns (string memory) {
450          return _name;
451      }
```

```
452
453     function symbol() public view returns (string memory) {
454         return _symbol;
455     }
```

```
456
457     function decimals() public view returns (uint8) {
458         return _decimals;
459     }
```

```
460
461     function totalSupply() public pure override returns (uint256) {
462         return _tTotal;
463     }
```

```
464
465     function balanceOf(address account) public view override returns (uint256) {
466         if (_isExcluded[account]) return _tOwned[account];
467         return tokenFromReflection(_rOwned[account]);
468     }
```

```
469
470     function transfer(address recipient, uint256 amount) public override returns (bool) {
471         _transfer(_msgSender(), recipient, amount);
472         return true;
473     }
```

```
474
475     function allowance(address owner, address spender) public view override returns (uint256) {
476         return _allowances[owner][spender];
477     }
```

```
478
479     function approve(address spender, uint256 amount) public override returns (bool) {
480         _approve(_msgSender(), spender, amount);
481         return true;
482     }
```

```
483
484     function transferFrom(address sender, address recipient, uint256 amount) public override returns (bool) {
485         _transfer(sender, recipient, amount);
486         _approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount, "ERC20: transfer amount exceeds allowance"));
487         return true;
488     }
```

```
494
495     function decreaseAllowance(address spender, uint256 subtractedValue) public virtual returns (bool) {
496         _approve(_msgSender(), spender, _allowances[_msgSender()][spender].sub(subtractedValue, "ERC20: decreased allowance below zero"));
497         return true;
498     }
```

```
499
500     function isExcluded(address account) public view returns (bool) {
501         return _isExcluded[account];
502     }
```

```
503
504     function totalFees() public view returns (uint256) {
505         return _tFeeTotal;
506     }
```

```
507
508     function reflect(uint256 tAmount) public {
509         address sender = _msgSender();
510         require(!_isExcluded[sender], "Excluded addresses cannot call this function");
511         (uint256 rAmount,,,,) = _getValues(tAmount);
512         _rOwned[sender] = _rOwned[sender].sub(rAmount);
513         _rTotal = _rTotal.sub(rAmount);
514         _tFeeTotal = _tFeeTotal.add(tAmount);
515     }
```

```
516
517     function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view returns(uint256) {
518         require(tAmount <= _tTotal, "Amount must be less than supply");
519         if (!deductTransferFee) {
520             (uint256 rAmount,,,,) = _getValues(tAmount);
521             return rAmount;
522         } else {
523             (,uint256 rTransferAmount,,,) = _getValues(tAmount);
524             return rTransferAmount;
525         }
526     }
```

Loop over unbounded data structure.
Gas consumption in function "includeAccount" in contract "TRDG" depends on the size of data structures or values that may grow unboundedly. If the data structure grows too large, the gas required to execute the code will exceed the block gas limit, effectively causing a denial-of-service condition. Consider that an attacker might attempt to cause this condition on purpose.

```
542
543     function includeAccount(address account) external onlyOwner() {
544         require(_isExcluded[account], "Account is already excluded");
545         for (uint256 i = 0; i < _excluded.length; i++) {
546             if (_excluded[i] == account) {
547                 _excluded[i] = _excluded[_excluded.length - 1];
548                 _tOwned[account] = 0;
549                 _isExcluded[account] = false;
550                 _excluded.pop();
551                 break;
552             }
553         }
554     }
```

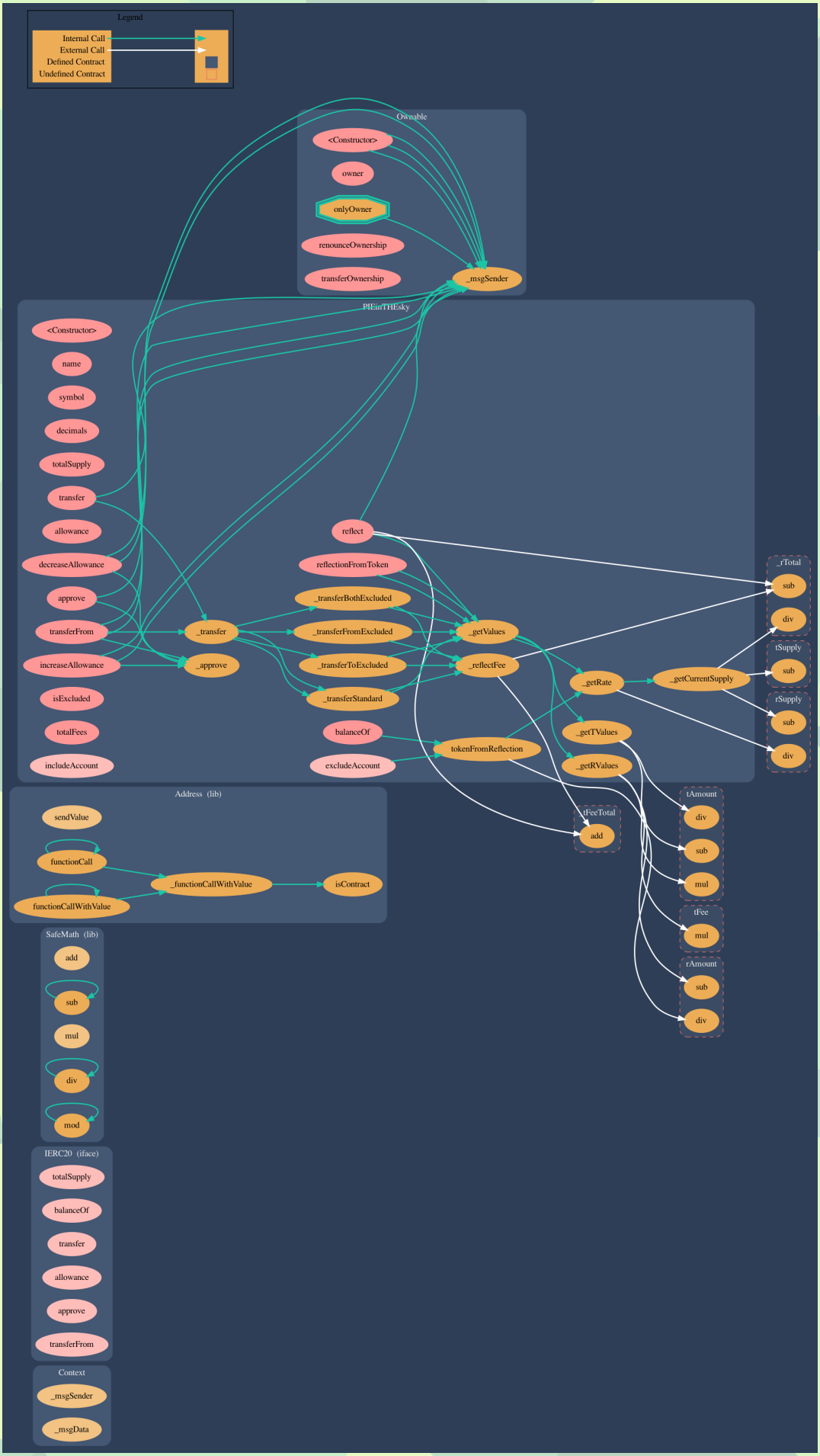
Loop over unbounded data structure.
Gas consumption in function "_getCurrentSupply" in contract "TRDG" depends on the size of data structures or values that may grow unboundedly. If the data structure grows too large, the gas required to execute the code will exceed the block gas limit, effectively causing a denial-of-service condition. Consider that an attacker might attempt to cause this condition on purpose.

```
646
647     function _getCurrentSupply() private view returns(uint256, uint256) {
648         uint256 rSupply = _rTotal;
649         uint256 tSupply = _tTotal;
650         for (uint256 i = 0; i < _excluded.length; i++) {
651             if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
652             rSupply = rSupply.sub(_rOwned[_excluded[i]]);
653             tSupply = tSupply.sub(_tOwned[_excluded[i]]);
654         }
655         if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
656         return (rSupply, tSupply);
657     }
658 }
659
```

LOW

A floating pragma is set.
The current pragma Solidity directive is ""^0.7.6"". 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.

```
5
6   // SPDX-License-Identifier: Unlicensed
7
8  pragma solidity ^0.7.6;
9
10 abstract contract Context {
11     function _msgSender() internal view virtual returns (address payable) {
12         return msg.sender;
13     }
14
```

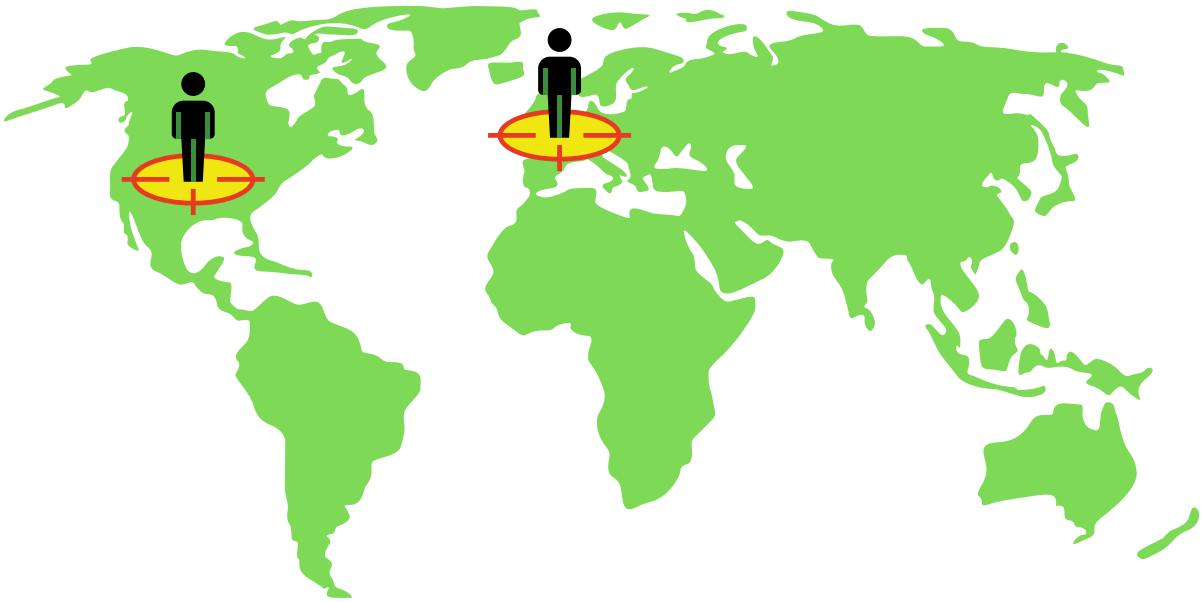

Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
<Constructor>
owner
onlyOwner
renounceOwnership
transferOwnership
_msgSender
PIEarTHEsky
<Constructor>
name
symbol
decimals
totalSupply
transfer
allowance
decreaseAllowance
approve
transferFrom
increaseAllowance
isExcluded
totalFees
includeAccount
reflect
reflectionFromToken
_transferBothExcluded
_transferFromExcluded
_transferToExcluded
_transferStandard
balanceOf
excludeAccount
_transfer
_approve
_getValues
_reflectFee
tokenFromReflection
_getRate
_getCurrentSupply
_getTValues
_getRValues
_rTotal
sub
div
tSupply
sub
rSupply
sub
div
tFeeTotal
add
tAmount
div
sub
mul
tFee
mul
rAmount
sub
div
Address (lib)
sendValue
functionCall
functionCallWithValue
_functionCallWithValue
isContract
SafeMath (lib)
add
sub
mul
div
mod
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData

Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	◆	NO !
Name	Public		NO !
Symbol	Public		NO !
Decimals	Public		NO !
TotalSupply	Public		NO !
BalanceOf	Public		NO !
Transfer	Public	◆	NO !
Allowance	Public		NO !
Approve	Public	◆	NO !
TransferFrom	Public	◆	NO !
IncreaseAllowance	Public	◆	NO !
DecreaseAllowance	Public	◆	NO !
IsExcluded	Public		NO !
TotalFees	Public		NO !
Reflect	Public	◆	NO !
ReflectionFromToken	Public		NO !
TokenFromReflection	Public		NO !
ExcludeAmount	External	◆	Only Owner
IncludeAmount	External	◆	Only Owner
_Approve	Private	◆	
_Transfer	Private	◆	
_TransferStandard	Private	◆	
_TransferToExcluded	Private	◆	
_TransferFromExcluded	Private	◆	
_TransferBothExcluded	Private	◆	
_ReflectFee	Private	◆	
_GetValues	Private		
_GetTValues	Private		
_GetRValues	Private		
_GetRate	Private		
_GetCurrentSupply	Private		

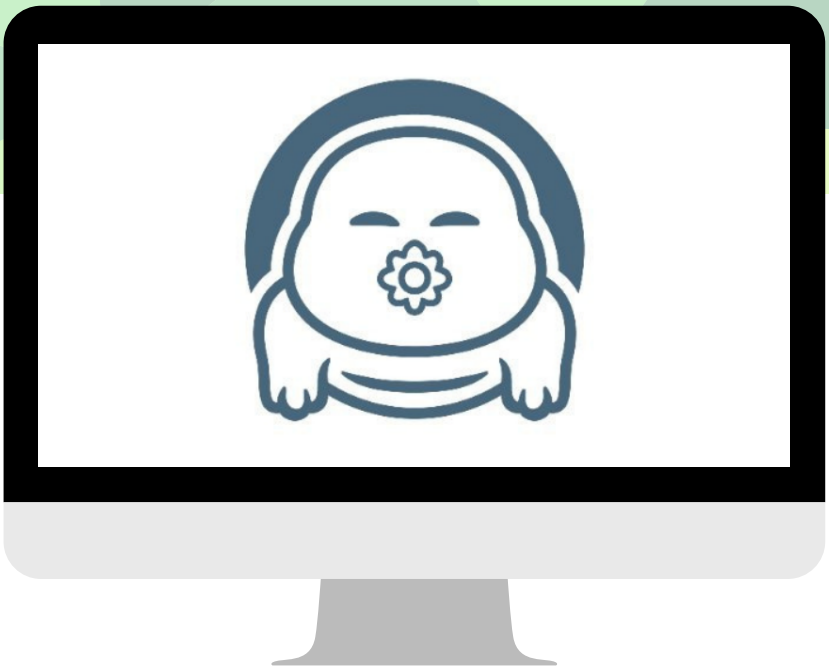
◆ = Function can modify state

LOCATION TEAM

TEAM USA AND FRANCE



SOCIAL MEDIA



https://twitter.com/pieintheskybsc?s=21

https://tardigrades.finance/

https://t.me/TardigradesOfficial
https://t.me/TardigradesAnnouncements

https://link.medium.com/VxpIRGLnogb

HTTPS://WWW.REDDIT.COM/R/TRDG_HODLERS/

NOTE AND CONCLUSION

The $TRDG's smart contract has no vulnerabilities that would jeopardise the interests of the project and its investors.

Some "micro weaknesses" are nevertheless present, but they do not indicate a need for modifications, and do not change the security of the investors.

As for the marketing and structure of the project, the progress of the project shows the professionalism and experience of the dev team.

TRDG is becoming and will become a crypto to follow closely in the coming months/years.

TEAM DÉCISION

TG : SPYONCRYPTO
TG CHAT: SPYONCRYPTO CHAT
TWITTER: @spyoncrypto
MAIL: SOCIAL@SPYONCRYPTO.COM
SITE WEB: WWW.SPYONCRYPTO.COM