Smart Contract Security Audit

Audit details:

Audited project:	Parallel Universe
Deployer address:	0x357B174d3690998845c0A5D3B2762E8c600BB814
Client contacts:	Parallel Universe team
Blockchain:	Binance Smart Chain
Project website:	https://www.pulproject.com

May, 2021
TechRate

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 below 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 investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate 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 TechRate 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, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, 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. No product code has been reviewed.

Background

TechRate was commissioned by Parallel Universe to perform an audit of smart contracts:

- *https://bscscan.com/address/0x11ed1d93bdd3ee180ca724ac82f911663f0dafb2#code*

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.

Contracts details

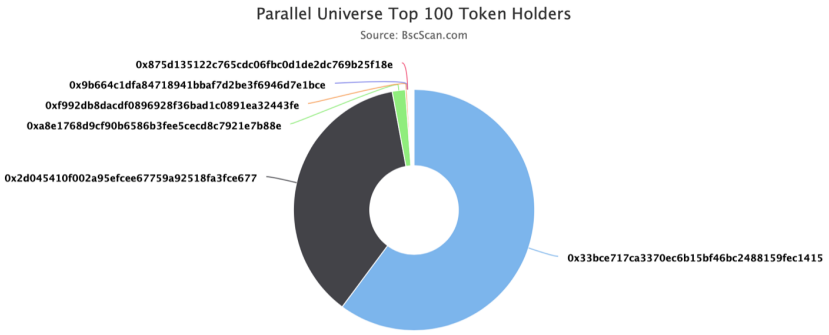
Token contract details for 16.05.2021.

Contract name:	Parallel Universe
Contract address:	0x11ed1d93bdD3Ee180CA724AC82F911663f0daFB2
Total supply:	995752906856624320288740
Token ticker:	PUL
Decimals:	9
Token holders:	1,567
Transactions count:	6,430
Top 100 holders dominance:	101.60%
Tax fee:	500
Total fees:	21042015030684345532541
Contract deployer address:	0x357B174d3690998845c0A5D3B2762E8c600BB814
Contract’s current owner address:	0x33bce717ca3370ec6b15bf46bc2488159fec1415

Parallel Universe token distribution

The top 100 holders collectively own 101.60% (1,011,651,961,137,840.00 Tokens) of Parallel Universe

Token Total Supply: 995,752,906,856,624.32 Token | Total Token Holders: 1,567

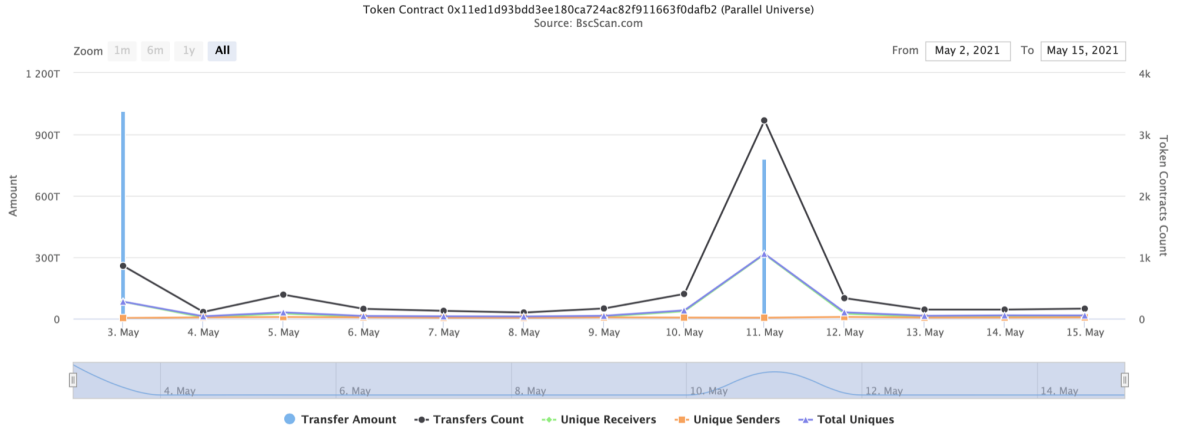


(A total of 1,011,651,961,137,840.00 tokens held by the top 100 accounts from the total supply of 995,752,906,856,624.32 token)

Parallel Universe contract interaction details

Time Series: Token Contract Overview

Mon 3, May 2021 - Sat 15, May 2021



Parallel Universe top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x33bce717ca3370ec6b15bf46bc2488159fec1415	608,821,521,564,383.850325209	61.1418%
2	0x2d045410f002a95efcee67759a92518fa3fce677	373,516,706,342,146.40418884	37.5110%
3	0xa8e1768d9cf90b6586b3fee5cecd8c7921e7b88e	17,827,736,480,318.274253129	1.7904%
4	0xf992db8dacdf0896928f36bad1c0891ea32443fe	2,258,225,541,064.203880639	0.2268%
5	0x9b664c1dfa84718941bbaf7d2be3f6946d7e1bce	521,147,952,940.418331173	0.0523%
6	0x875d135122c765cdc06fbc0d1de2dc769b25f18e	516,625,552,455.727447461	0.0519%
7	0xe70039282c64f6f770062162fbc5e2f63d7d9ce9	466,126,435,293.948806701	0.0468%
8	0x7c9cdc1bc5f1152820744fd0b4a8ef46a6671aa9	460,018,901,271.992503568	0.0462%
9	0x47d65bd572fe41dc360892906b2fe014afb31d44	392,230,858,316.005124006	0.0394%
10	0x7a3d0627fb717261366e23ed7fe34d3efe97e003	377,652,083,255.648868667	0.0379%

Contract functions details

+ Context
  - [Int] _msgSender
  - [Int] _msgData
+ [Int] IBEP20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #
+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div
  - [Int] mod
  - [Int] mod
+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Prv] _functionCallWithValue #
+ Ownable (Context)
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
+ CoinToken (Context, IBEP20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #

- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Pub] isExcluded
- [Pub] isCharity
- [Pub] totalFees
- [Pub] totalBurn
- [Pub] totalCharity
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Ext] excludeAccount #
  - modifiers: onlyOwner
- [Ext] includeAccount #
  - modifiers: onlyOwner
- [Ext] setAsCharityAccount #
  - modifiers: onlyOwner
- [Pub] burn #
- [Pub] updateFee #
  - modifiers: onlyOwner
- [Int] _burn #
- [Pub] mint #
  - modifiers: onlyOwner
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] _transferStandard #
- [Prv] _standardTransferContent #
- [Prv] _transferToExcluded #
- [Prv] _excludedFromTransferContent #
- [Prv] _transferFromExcluded #
- [Prv] _excludedToTransferContent #
- [Prv] _transferBothExcluded #
- [Prv] _bothTransferContent #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTBasics
- [Prv] getTTransferAmount
- [Prv] _getRBasics
- [Prv] _getRTransferAmount
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _sendToCharity #
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Prv] _getTaxFee

($) = payable function
= non-constant function

Issues Checking Status

№	Issue description.	Checking status
1	Compiler errors.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Low issues
10	Methods execution permissions.	Passed
11	Economy model of the contract.	High issues
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Open Zeppelin contracts implementation and usage.	Passed
21	Fallback function security.	Passed

Security Issues

High Severity Issues

1. Wrong burn and mint

Issue:

In burn and mint functions there are wrong values adding because of not converting _value. _rOwner and _tTotal show balances in different modes and same values will be added / subtracted to them, which will make it wrong.

```
function _burn(address _who⬆, uint256 _value⬆) internal {
    require(_value⬆ <= _rOwned[_who⬆]);
    _rOwned[_who⬆] = _rOwned[_who⬆].sub(_value⬆);
    _tTotal = _tTotal.sub(_value⬆);
    emit Transfer(_who⬆, address(0), _value⬆);
}

function mint(address account⬆, uint256 amount⬆) onlyOwner() public {

    _tTotal = _tTotal.add(amount⬆);
    _rOwned[account⬆] = _rOwned[account⬆].add(amount⬆);
    emit Transfer(address(0), account⬆, amount⬆);
}
```

Recommendation:

Please check if the addresses are included in reward or not and add the values correctly by multiplying by the rate.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- ❏ The function `includeAccount()` uses the loop to find and remove addresses from the `_excluded` list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.

```
function includeInReward(address account⬆) external onlyOwner() {
    require(_isExcluded[account⬆], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account⬆) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account⬆] = 0;
            _isExcluded[account⬆] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- ❏ The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:

Use EnumerableSet instead of array or do not use long arrays.

Owner privileges (In the period when the owner is not renounced)

- ❏ Owner can exclude from the fee.

```
function excludeAccount(address account) external onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

- ❏ Owner can change charity address

```
function setAsCharityAccount(address account) external onlyOwner() {
    require(!_isCharity[account], "Account is already charity account");
    _isCharity[account] = true;
    FeeAddress = account;
}
```

- ❏ Owner can mint

```
function mint(address account, uint256 amount) onlyOwner() public {

    _tTotal = _tTotal.add(amount);
    _rOwned[account] = _rOwned[account].add(amount);
    emit Transfer(address(0), account, amount);
}
```

- ❏ Owner can change fees

```
function updateFee(uint256 _txFee,uint256 _burnFee,uint256 _charityFee) onlyOwner() public{
    _TAX_FEE = _txFee* 100;
    _BURN_FEE = _burnFee * 100;
    _CHARITY_FEE = _charityFee* 100;
    ORIG_TAX_FEE = _TAX_FEE;
    ORIG_BURN_FEE = _BURN_FEE;
    ORIG_CHARITY_FEE = _CHARITY_FEE;
}
```

Conclusion

Smart contracts contain high severity issues. LP pair contract is not checked.

Liquidity locking details provided by the team:
https://unicrypt.network/amm/pancakev2/pair/0xf992db8dacdf0896928f36bad1c0891ea32443fe

Techrate note:
Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.