



Smart Contract Security Audit Report

Victoria VR

April 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

Victoria VR



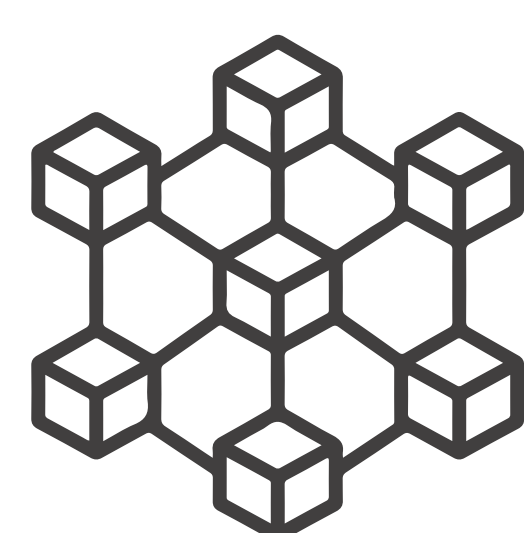
Deployer address

0x3c4172ffFCAeFBfed90C110c1e1fe63B0224e13A



Client contacts

Victoria VR team



Blockchain

Ethereum



Website

www.victoriavr.com

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) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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

HeckSafe was commissioned by Victoria VR to perform an audit of smart contracts:

- <https://etherscan.io/address/0x7d5121505149065b562c789a0145ed750e6e8cdd#code>

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issue 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

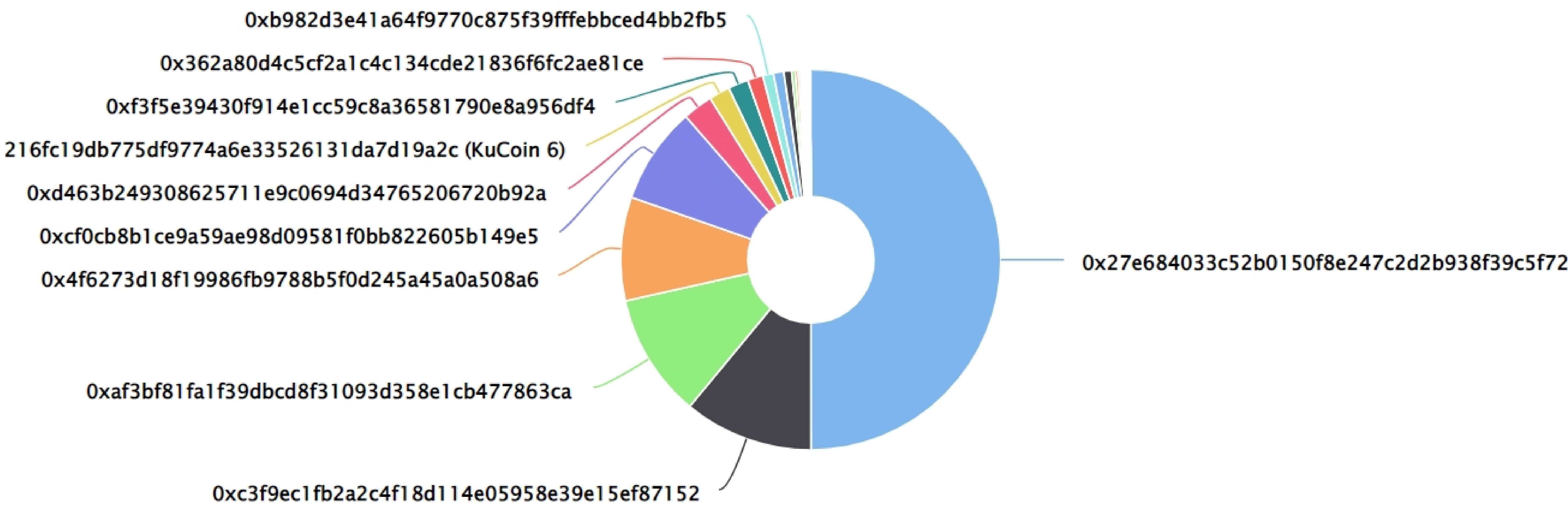
Token contract details for 13.04.2022

Contract name	: Victoria VR
Contract address	: 0x7d5121505149065b562C789A0145eD750e6E8cdD
Total supply	: 16,800,000,000
Token Ticker	: VR
Decimals	: 18
Network	: Ethereum
Transactions count	: 41,324
Token Holders	: 7126 addresses
Contract deployer address	: 0x3c4172ffFCAeFBfed90C110c1e1fe63B0224e13A
Owner address	: 0xeb4F2E0BeB5c518eabF7808AA9b1724aAeE244Bb

Victoria VR Token Distribution

Victoria VR Top 500 Token Holders

Source: Etherscan.io



Victoria VR Top 10 Token Holders

(A total of 16,270,117,358.15 tokens held by the top 10 accounts from the total supply of 16,800,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x27e684033c52b0150f8e247c2d2b938f39c5f721	8,400,000,000	50.0000%
2	0xc3f9ec1fb2a2c4f18d114e05958e39e15ef87152	1,848,000,000	11.0000%
3	0xaf3bf81fa1f39dbcd8f31093d358e1cb477863ca	1,769,491,524	10.5327%
4	0x4f6273d18f19986fb9788b5f0d245a45a0a508a6	1,478,400,000	8.8000%
5	0xcf0cb8b1ce9a59ae98d09581f0bb822605b149e5	1,385,304,407	8.2459%
6	0xd463b249308625711e9c0694d34765206720b92a	427,681,321	2.5457%
7	KuCoin 6	298,949,400	1.7795%
8	0xf3f5e39430f914e1cc59c8a36581790e8a956df4	289,800,000	1.7250%
9	0x362a80d4c5cf2a1c4c134cde21836f6fc2ae81ce	219,948,032.69706421883420999	1.3092%
10	0xb982d3e41a64f9770c875f39fffebbced4bb2fb5	152,542,673.45762712	0.9080%

Contract functions details

VRToken.sol

+ VictoriaVR (Ownable, ERC20Burnable, ERC20Pausable, ERC20Snapshot)

- [Pub] <constructor> #
- [Int] _beforeTokenTransfer#
- [Pub] pause
 - Modifier: onlyOwner, whenNotPaused
- [Pub] unpause
 - Modifier: onlyOwner, whenNotPaused
- [Pub] snapshot
 - Modifier: onlyOwner

Address.sol

+ [Lib] Address

- [Int] isContract
- [Int] sendValue
- [Int] functionCall
- [Int] functionCall
- [Int] functionCallWithValue
- [Int] functionCallWithValue
- [Pvt] _functionCallWithValue

Arrays.sol

+ [Lib] Arrays

- [Int] findUpperBound

Context.sol

+ Context

- [Int] _msgSender
- [Int] _msgData

Counters.sol

+ [Lib] Counters

- [Int] current
- [Int] increment
- [Int] decrement

Contract functions details

ERC20.sol

+ ERC20 (Context, IERC20)

- [Pub] <constructor>
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer #
- [Pub] allowance
- [Pub] approve #
- [Pub] transferFrom #
- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Int] _transfer #
- [Int] _mint #
- [Int] _burn #
- [Int] _approve #
- [Int] _setupDecimals
- [Int] _beforeTokenTransfer

ERC20Burnable.sol

+ ERC20Burnable (Context, ERC20)

- [Pub] burn #
- [Pub] burnFrom #

ERC20Pausable.sol

+ ERC20Pausable (ERC20, Pausable)

- [Int] _beforeTokenTransfer #

ERC20Snapshot.sol

+ ERC20Snapshot (ERC20)

- [Int] _snapshot
- [Pub] balanceOfAt
- [Pub] totalSupplyAt
- [Int] _beforeTokenTransfer
- [Pvt] _valueAt
- [Pvt] _updateAccountSnapshot

Contract functions details

-[Pvt] _updateTotalSupplySnapshot

-[Pvt] _updateSnapshot

-[Pvt] _lastSnapshotId

IERC20.sol

+ [Int] IERC20

-[Ext] totalSupply

-[Ext] balanceOf

-[Ext] transfer #

-[Ext] allowance

-[Ext] approve #

-[Ext] transferFrom #

Math.sol

+ [Lib] Math

-[Int] max

-[Int] min

-[int] average

Ownable.sol

+ Ownable (Context)

-[Int] <constructor>

-[Pub] owner

-[Pub] renounceOwnership

-Modifier: onlyOwner

-[Pub] transferOwnership

-Modifier: onlyOwner

Pausable.sol

+ Pausable (Context)

-[Int] <constructor>

-[Pub] paused

-[Int] _pause

-Modifier: whenNotPaused

-[Int] _unpause

-Modifier: whenPaused

Contract functions details

```
SafeMath.sol
+[Lib] SafeMath
  -[Int] add
  -[Int] sub
  -[Int] sub
  -[Int] mul
  -[Int] div
  -[Int] div
  -[Int] mod
  -[Int] mod
```

(\$) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Low issue
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

✔ Critical Severity Issues

No critical severity issue found.

✔ High Severity Issues

No high severity issue found.

✔ Medium Severity Issues

No Medium severity issue found.

✔ Low Severity Issues

Two low severity issue found.

1. Unlocked Compiler Version.

- **Description**

The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version v0.6.0 the contract should contain the following line:

```
pragma solidity 0.6.0;
```

2. Scoping and Declarations.

Unused function.

- **Description**

The `_msgData`, `_mint`, `decrement` function does nothing.

- **Location**

Context.sol -> `_msgData` function

ERC20.sol -> `_mint` function

Counters.sol -> `decrement` function

- **Recommendation**

We advise to remove unused code.

Owner Privileges

Owner Privileges (in the period when the owner is not renounced) :

- Victoria VR Contract:
 - Owner can pause tokens transfes.
 - Owner can unpause tokens transfers.
 - Owner can snapshot.
 - Owner can change ownership.
 - Owner can renounce ownership.

Conclusion

Smart contract contains low severity issues!

HackSafe 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.