



Smart contracts security assessment

Final report

[Tariff: Standard](#)

XRP20 Token

August 2023



0xguard.com



hello@0xguard.com

Contents

1. Introduction	3
2. Contracts checked	3
3. Procedure	3
4. Known vulnerabilities checked	4
5. Classification of issue severity	5
6. Issues	5
7. Conclusion	7
8. Disclaimer	8

Introduction

The report has been prepared for **XRP20 Token**.

Audited token is an ERC-20 standard token. XRP20 token doesn't have mint functionality and all transfers are subjected to 0.1% burn with a whitelist, governed by the contract's owner. Tax is not applied for transferring amounts below 1000 wei threshold.

The code in the @XRP20Dev/XRP20TokenContract Github repo was audited in the [4c6271b](#) commit.

Name	XRP20 Token
Audit date	2023-08-14 - 2023-08-16
Language	Solidity
Platform	Ethereum

Contracts checked

Name	Address
XRP20Token	

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
<u>Unencrypted Private Data On-Chain</u>	passed
<u>Code With No Effects</u>	passed
<u>Message call with hardcoded gas amount</u>	passed
<u>Typographical Error</u>	passed
<u>DoS With Block Gas Limit</u>	passed
<u>Presence of unused variables</u>	passed
<u>Incorrect Inheritance Order</u>	passed
<u>Requirement Violation</u>	passed
<u>Weak Sources of Randomness from Chain Attributes</u>	passed
<u>Shadowing State Variables</u>	passed
<u>Incorrect Constructor Name</u>	passed
<u>Block values as a proxy for time</u>	passed
<u>Authorization through tx.origin</u>	passed
<u>DoS with Failed Call</u>	passed
<u>Delegatecall to Untrusted Callee</u>	passed
<u>Use of Deprecated Solidity Functions</u>	passed
<u>Assert Violation</u>	passed
<u>State Variable Default Visibility</u>	passed
<u>Reentrancy</u>	passed

<u>Unprotected SELFDESTRUCT Instruction</u>	passed
<u>Unprotected Ether Withdrawal</u>	passed
<u>Unchecked Call Return Value</u>	passed
<u>Floating Pragma</u>	passed
<u>Outdated Compiler Version</u>	passed
<u>Integer Overflow and Underflow</u>	passed
<u>Function Default Visibility</u>	passed

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

No issues were found

Medium severity issues

No issues were found

Low severity issues

1. Standard violation (XRP20Token)

Status: Open

The ERC-20 token standard [requires](#) transfers with zero amount to be processed. The XRP20 token reverts all such transfers explicitly.

```
function _transfer(address sender, address recipient, uint256 amount) internal
virtual {
    require(amount > 0, 'ERC20: transfer amount zero');
    require(sender != address(0), 'ERC20: transfer from the zero address');
    require(recipient != address(0), 'ERC20: transfer to the zero address');

    uint256 senderBalance = _balances[sender];
    require(senderBalance >= amount, 'ERC20: transfer amount exceeds balance');
    unchecked {
        _balances[sender] = senderBalance - amount;
    }
    _balances[recipient] += amount;

    emit Transfer(sender, recipient, amount);
}
```

Recommendation: Comply to the standard.

XRP20 team comment: The XRP20 Token reverts all such transfers explicitly, this function has been added to protect token holders from Zero-Value Token Transfer Attacks. See more information here <https://info.etherscan.com/zero-value-token-transfer-attack/>.

Conclusion

XRP20 Token XRP20Token contract was audited. 1 low severity issue was found.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without 0xGuard prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts 0xGuard to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.



 Guard