HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: XP Network
Date: October 14th, 2021

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

Document

Name	Smart Contract Code Review and Security Analysis Report for XP Network.
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	BEP20 token; Staking; ERC721 token; ERC1155 token; NFT Minter
Platform	Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/xp-network/staker https://github.com/xp-network/XP.network-HECO-Migration
Commit	3b03344992ad28cce3cde4e808d67dc746fdb0bc a97922dd0204d6044673a0f1eb884e7e1335d431
Technical Documentation	NO
JS tests	YES
Website	
Timeline	05 OCTOBER 2021 - 14 OCTOBER 2021
Changelog	13 OCTOBER 2021 - INITIAL AUDIT 14 OCTOBER 2021 - SECOND REVIEW

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Table of contents

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Introduction

Hacken OÜ (Consultant) was contracted by XP Network (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contract and its code review conducted between October 5th, 2021 - October 13th, 2021.

The second review conducted on October 14th, 2021.

Scope

The scope of the project is smart contracts in the repository:

Repository 1:
https://github.com/xp-network/staker
Commit 1:
3b03344992ad28cce3cde4e808d67dc746fdb0bc
Repository 2:
https://github.com/xp-network/XP.network-HECO-Migration
Commit 2:
a97922dd0204d6044673a0f1eb884e7e1335d431
Technical Documentation: No
JS tests 1: Yes (https://github.com/xp-network/staker/tree/3b03344992ad28cce3cde4e808d67dc746fdb0bc/test)
JS tests 2: (https://github.com/xp-network/XP.network-HECO-Migration/tree/a97922dd0204d6044673a0f1eb884e7e1335d431/web3-mint-contract/test)
Contracts 1:
XpNet.sol
XpNetStaker.sol
Contracts 2:
Minter.sol
UserNftMinter.sol
XPNet.sol
XPNft.sol

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	▪ Reentrancy ▪ Ownership Takeover ▪ Timestamp Dependence ▪ Gas Limit and Loops ▪ DoS with (Unexpected) Throw ▪ DoS with Block Gas Limit ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ Costly Loop ▪ ERC20 API violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency ▪ Data Consistency
Functional review	▪ Business Logics Review ▪ Functionality Checks ▪ Access Control & Authorization ▪ Escrow manipulation ▪ Token Supply manipulation ▪ Assets integrity ▪ User Balances manipulation ▪ Data Consistency manipulation ▪ Kill-Switch Mechanism ▪ Operation Trails & Event Generation

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Executive Summary

According to the assessment, the Customer's smart contracts are well-secured.

Insecure	Poor secured	Secured	Well-secured

You are here 

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview section.

As a result of the audit, security engineers found **2** medium and **7** low severity issues.

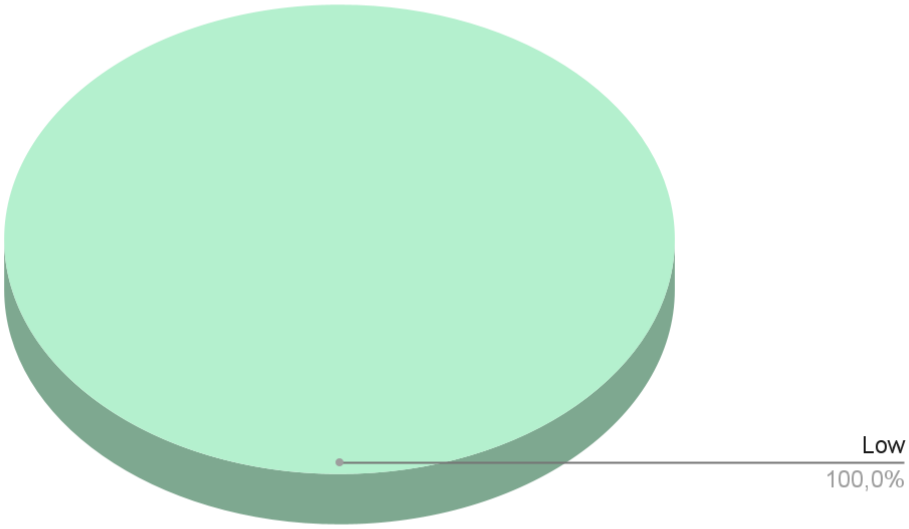
After the second review security engineers found **1** low severity issue.

Notice:

The XpNetStaker could have reentrancy on the “withdraw” and “withdrawRewards” functions in the case of using a token with the callback on receive as the primary token. But while in the tests we saw that the XpNet token is used as the primary and it doesn’t contain the callback, it is good. But please keep in mind that using any ERC20 token with the callback to the receiver will cause the issue.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Graph 1. The distribution of vulnerabilities after the audit.



Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

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

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Audit overview

Critical

No critical issues were found.

High

No high issues were found

Medium

1. Tests are failing

Scope: Staker

There are 21 test cases provided with the code, but 19 of them are failing. While no documentation is provided there could be a business logic mismatch between the code and tests.

Recommendation: Please make sure the entire business logic is covered by the successful running tests.

Status: Fixed.

2. Tests are failing

Scope: Minter

There are 9 test cases provided with the code, but 2 of them (withdraw nft and transfer to foreign) are failing. While no documentation is provided there could be a business logic mismatch between the code and tests.

Recommendation: Please make sure the entire business logic is covered by the successful running tests

Status: Fixed.

Low

1. Too many digits

Literals with many digits are difficult to read and review.

Scope: Staker

Contracts: XpNet, XpNetStaker

Functions:
- XpNet constructor at line: #11
- XpNetStaker stake at lines: #97, 99

Recommendation: Please consider using scientific notation or/and ether units to declare numbers (ie: *5e9 ether*). Also, instead of multiplying

by “(10**18)” it’s better to use just a keyword “**ether**”, which actually does the same but is better to read/understand (ie: “1_500 ether” instead of “1_500 * (10**18)”).

Status: Fixed.

2. State variable could be declared as immutable

A state variable that is initialized in the constructor and never changes its value should be declared immutable to save gas.

Scope: Staker

Contracts: XpNetStaker

Lines: #23

Recommendation: Please add the keyword immutable to the state variable to save gas.

Status: Fixed.

3. State variable could be declared as immutable

A state variable that is initialized in the constructor and never changes its value should be declared immutable to save gas.

Scope: Minter

Contracts: Minter

Lines: #21-22

Recommendation: Please add the keyword immutable to the state variable to save gas.

Status: Fixed.

4. No event on adding/deducting tokens

There are two functions in the contract (sudoAddToken and sudoDeductToken) which are adding and deducting tokens to the stakeholders. But those functions don’t emit events which makes it harder to track such changes off-chain.

Scope: Staker

Contracts: XpNetStaker

Functions: sudoAddToken, sudoDeductToken

Recommendation: Please emit events on making corrections to the stakeholders.

Status: Fixed.

5. Conformance to Solidity naming conventions

Solidity defines a naming convention that should be followed.

Scope: Minter

Contracts: Minter

Functions: freeze_erc721, withdraw_nft, withdraw, validate_withdraw_fees, validate_set_threshold, validate_unpause_bridge, validate_pause_bridge, validate_remove_validator, validate_add_validator, validate_whitelist_nft, validate_unfreeze_nft, validate_unfreeze, validate_transfer_nft, validate_transfer, validate_action

Recommendation: Please follow a naming convention.

6. A public function that could be declared external

public functions that are never called by the contract should be declared **external** to save gas.

Scope: Staker

Contracts: XpNetStaker

Functions: setBaseUri, stake, withdraw, withdrawRewards, checkIsUnlocked, showAvailableRewards, sudoAddToken, sudoDeductToken, sudoWithdrawToken

Recommendation: Use the **external** attribute for functions never called from the contract.

Status: Fixed.

7. A public function that could be declared external

public functions that are never called by the contract should be declared **external** to save gas.

Scope: Minter

Contracts: Minter

Functions: freeze, freeze_erc721, withdraw_nft, withdraw, validate_withdraw_fees, validate_set_threshold, validate_unpause_bridge, validate_pause_bridge, validate_remove_validator, validate_add_validator, validate_whitelist_nft, validate_unfreeze_nft, validate_unfreeze, validate_transfer_nft, validate_transfer, validate_action

Recommendation: Use the **external** attribute for functions never called from the contract.

Status: Fixed.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

As a result of the audit, security engineers found **2** medium and **7** low severity issues.

After the second review security engineers found **1** low severity issue.

Notice:

The XpNetStaker could have reentrancy on the “withdraw” and “withdrawRewards” functions in the case of using a token with the callback on receive as the primary token. But while in the tests we saw that the XpNet token is used as the primary and it doesn’t contain the callback, it is good. But please keep in mind that using any ERC20 token with the callback to the receiver will cause the issue.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only – we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.