QuillAudits

Audit Report
April, 2023

For

xterio

Table of Content

Executive Summary

Project Name	Xterio
Overview	Xterio Tech is a Play-and -earn gaming platform and a GameFi-as-a-service (GaaS) solution that will serve as an onboarding catalyst to web3. Xterio will deploy titles for competitive gamers worldwide and leverage its extensive global network and experience building games to develop, finance, acquire, publish and distribute games in partnership with best-in-class studios. Xterio focuses its partner studios on building world-class Play-and-Earn games on its Xterio platform sharing common universes.
Method	Manual Review, Functional Testing, Automated Testing etc.
Scope of Audit	The scope of this audit was to analyze the changes in Xterio tech smart contract's codebase for quality, security, and correctness. **Github:** *https://github.com/XterioTech/smart-contracts*
Commit	4c9c9a03d2291ac127fb3db7ea81a7c0b7c5e7cc
Fixed In	1f06dc7ccb6ba1d479c0c8479499747eb9e4708c

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	0
Partially Resolved Issues	0	0	0	0
Resolved Issues	1	0	1	0

Types of Severities

High

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

Checked Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- BEP20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis.

Manual Testing

High Severity Issues

1. Malicious user can replay the signatures.

Description

At Badge.sol#L155 and XSoul.sol#L84, In claimBadge and claimXSoul function respectively. Offchain computed signature is provided that does not have any expiry or nonce which would allow the recipient to re-use the same signature to mint the badge or a xSoul, While it would be a success only when the malicious user's badge get revoked then user is free to re-use the same signature to mint the new badge. Similarly for xSoul, If it get removed or burned from user wallet then user can re-use the same signature to mint the xSoul.

Recommendation

We recommend using the nonce or expiry during the generation of the signature so it can not be re-used again.

Status

Resolved

Medium Severity Issues

No issues were found

Low Severity Issues

2. Missing sanity check on expiry during setting user.

Description

At ERC4907.sol#L38, setUser function is used to set the user with given expiry value. However, In current implementation expiry value does not get validated before setting the user. Because of this an expired user can be added which causes confusion at offchain indexers when those indexers curates UpdateUser event

Recommendation

We recommend to adding a sanity check for expires param to make sure that it is greater than block.timestamp.

Status

Resolved

Informational Issues

No issues were found

Functional Testing

The complete functional testing report has been attached below: Xterio test case

Some of the tests performed are mentioned below:

- should be able to mint ERC721 in batch.
- should be able to mint ERC1155 in batch.
- should be able to set URI to ERC721 and ERC1155.
- should be able to mint ERC20 token.
- should be able to mint capped ERC20 token.
- should be able to safe transfer the ERC721.
- should be able to safe transfer the ERC1155.
- should be able to set user.
- should be able to create user record.
- should be able to delete user record.
- should be able to issue badge by operator.
- should be able to claim badge.
- should be able to revoke badge.
- should be able to set minimum amount for redeem.
- should be able to redeem XSoul.
- should be able to claimXSoul.
- should be able to revokeXSoul.
- should be able to issueXSoul.
- should be able to set redeem checker.

Closing Summary

One issue of High severity and one issue of low severity was found which are now ready to be fixed by developers. Some suggestions and best practices are also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits smart contract audit is not a security warranty, investment advice, or an endorsement of the Xterio Tech team. This audit does not provide a security or correctness guarantee for the audited smart contracts.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multi-step process. One audit cannot be considered enough. We recommend that Xterio Tech put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

700+
Audits Completed

$16B
Secured

700K
Lines of Code Audited

Follow Our Journey

Audit Report
April, 2023

For

xterio

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com