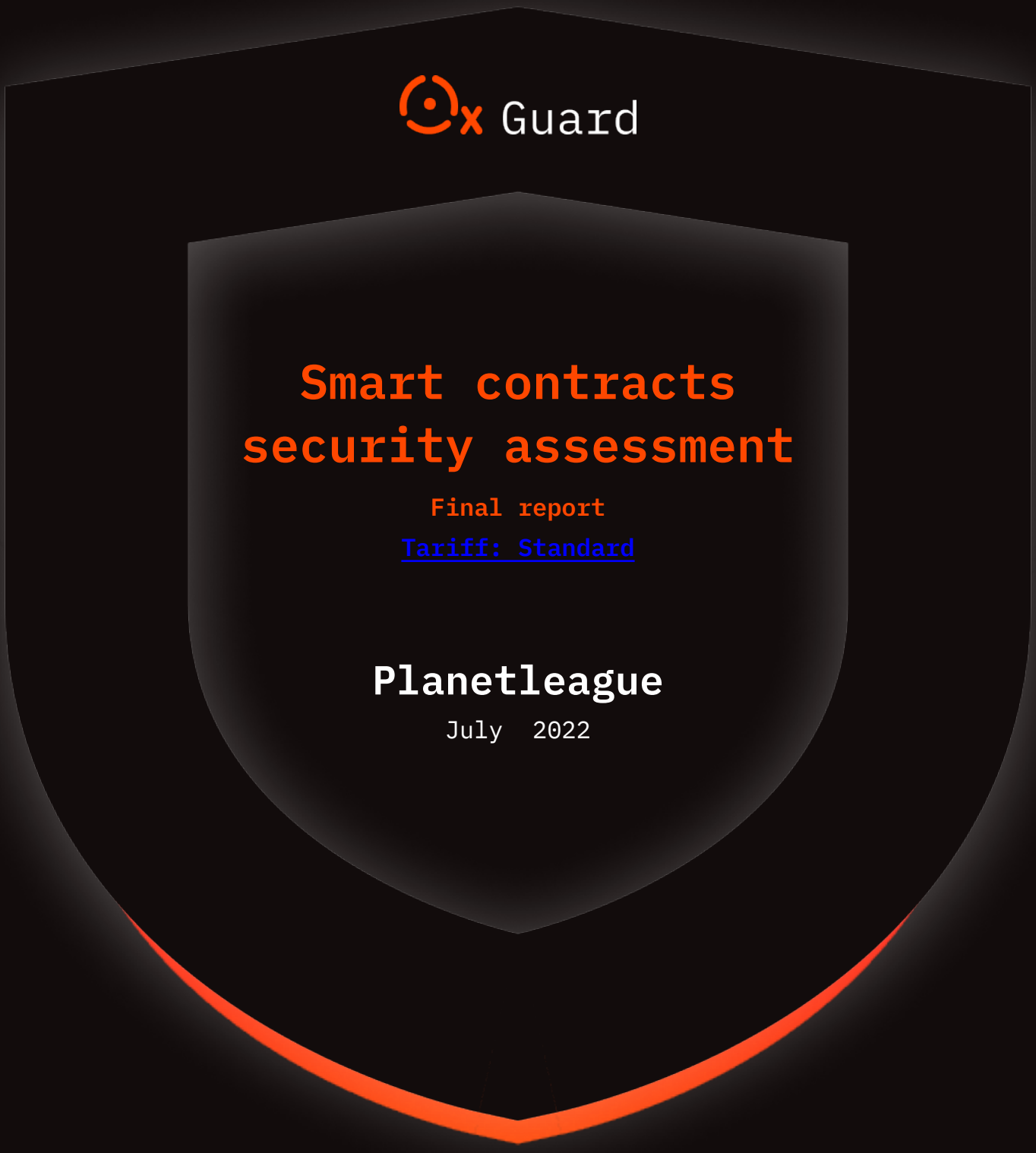

Guard

Smart contracts security assessment

Final report

Tariff: Standard

Planetleague

July 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for Planetleague. Three contracts were audited: PLS, PLSGame and PLSBonus. PLSGame is a contract for games: two players invest the same amount of funds and one of them wins all the invested assets of this game. The code is available in the Github repository. The code was checked in the baabab7 commit and rechecked in 060bb06.

Name	Planetleague
Audit date	2022-06-09 - 2022-07-13
Language	Solidity
Platform	Harmony

Contracts checked

Name	Address
PLS	https://github.com/jcauvet/plg-pls-contracts/blob/060bb068a8e2e8dccf1751097c6f82ccf505b67e/contracts/PLS.sol
PLSGame	https://github.com/jcauvet/plg-pls-contracts/blob/060bb068a8e2e8dccf1751097c6f82ccf505b67e/contracts/PLSGame.sol
PLSBonus	https://github.com/jcauvet/plg-pls-contracts/blob/060bb068a8e2e8dccf1751097c6f82ccf505b67e/contracts/PLS_Bonus.sol

Multiple contracts

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
Unencrypted Private Data On-Chain	passed
Code With No Effects	passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	passed
Presence of unused variables	passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed
Shadowing State Variables	passed
Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed

Use of Deprecated Solidity Functions	passed
Assert Violation	passed
State Variable Default Visibility	passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed
Floating Pragma	passed
Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	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

1. 100% fee - FIXED (PLS)

In the `setTaxPercent()` function, there is no limit on the `_taxPerc` parameter, so the owner can set a 100% fee in transfers.

Recommendation: It is recommended to limit the `_taxPerc` parameter.

2. 100% fee - FIXED (PLSGame)

In the `setPlatfee()` function, there is no limit on the `_platFee` parameter, so the owner can set a 100% fee at the end of the game. In such a case, the winner of the game receives 0 tokens from their victory.

Recommendation: It is recommended to limit the `_platFee` parameter so that it does not exceed 40%, otherwise the winner receives 0 tokens from the victory.

3. The winner is chosen by the owner of the contract (PLSGame)

With the `markGameComplete()` function, the owner of the contract can choose who won the game with a particular `gameId`.

Team response: The Planet League platform is designed to adjudicate real world esports and other competitive games and decide on the outcome based on evidence submitted by users, hence the contract is designed to provide the platform this level of control.

Medium severity issues

1. Unsafe function (PLSBonus)

Using the `withdrawToken()` and `withdrawAllToken()` functions, the owner can withdraw all assets from the contract address, including bonus tokens.

Team response: The withdraw token control pertaining to the bonus contract are for free (bonus)

tokens that users are rewarded based on their engagement with the platform. The withdraw token function in the contract has been designed in this manner intentionally for the following reasons: a) if users inadvertently deposit tokens into this contract and request our help to withdraw, and, b) if the bonus tokens issued to users expire and hence need to be withdrawn.

Low severity issues

1. Lacks validation of input parameters - FIXED (PLS)

The functions `setTaxWallet()` and `whitelistAddress()` does not check the input addresses against a null address.

2. Gas optimization (PLS)

All public functions can be declared as `external` to save gas.

Team response: Gas fees can vary significantly on each blockchain and the platform reserves the right to set who absorbs these gas charges to ensure economic viability of the platform.

3. Variable must be immutable - FIXED (PLSGame)

Variable `token` on 33L must be `immutable`.

4. Lacks validation of input parameters - FIXED (PLSGame)

The contract constructor, `setFeeWallet()` and `setBonusAddress()` does not check the input addresses against a null address.

5. Gas optimization - FIXED (PLSBonus)

The `setPlsGameAdd()` function can be declared as external to save gas.

6. Variable must be immutable - FIXED (PLSBonus)

The variable `bonusToken` in 15L must be `immutable`.

7. Floating Pragma - FIXED (Multiple contracts)

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Recommendation: Lock the pragma version and also consider known bugs (link) for the compiler version that is chosen.

8. Few events (Multiple contracts)

Many set functions from the contracts lack events.

Team response: To improve efficiency and control gas fee events, the platform tracks events on the application side.

Conclusion

Planetleague PLS, PLSGame, PLSBonus, Multiple contracts contracts were audited. 3 high, 1 medium, 8 low severity issues were 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.

0x Guard