SOLIDIFIED

Audit Report for Xaya - March 19, 2022

Summary

Audit Report prepared by Solidified covering the Xaya accounts and policy smart contracts.

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on 14 March 2022.

Audited Files

The source code has been supplied in the form of two GitLab repositories:

https://github.com/xaya/polygon-contract

Commit number: `6271ae0ec19432b74e3b31be1bf5cadc26dc9793`

The scope of the audit was limited to the following files:

```
contracts
|-- Utf8.sol
|-- XayaAccounts.sol
`-- XayaPolicy.sol
```

The remaining contracts have been excluded specifically from the scope of the audit.

Intended Behavior

The smart contracts implement an ERC-721-based name service.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does equate to a higher risk. Certain bugs are more easily detected in unit testing than in a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Low	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

SOLIDIFIED

Audit Report for Xaya - March 19, 2022

Issues Found

Solidified found that the Xaya contracts contain no critical issues, no major issues, 1 minor issue, in addition to 1 informational note.

We recommend all issues are amended, while the notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	Dependencies with well-known security vulnerabilities in the build system	Minor	Resolved
2	XayaPolicy.sol: Consider using 0x7e as upper limit	Note	Resolved

SOLIDIFIED

Audit Report for Xaya - March 19, 2022

Critical Issues

No critical issues have been identified

Major Issues

No critical issues have been identified

Minor Issues

1. Dependencies with well-known security vulnerabilities in the build system

The build and test system relies on a number of outdated JavaScript libraries with well-known security vulnerabilities, some of which are critical. Since these are only used for deployment and testing they do not constitute a smart contract security risk. However, outdated dependencies in the build system make operational security incidents, such as key leakage more likely.

Recommendation
Consider using `npm audit` to identify vulnerable dependencies and update them.

Informational Notes

2. XayaPolicy.sol: Consider using 0x7e as upper limit

The `checkMove` method allows bytes from 0x20 to 0x7f to be part of the move string. Since the byte `0x7f` represents control, it is recommended to not allow this as a valid move byte.

```
166| require (mvBytes[i] >= 0x20 && mvBytes[i] < 0x80, "invalid move
   | data");
```

Recommendation
Consider adding a validation to not include 0x7f as part of the move string.

Audit Report for Xaya - March 19, 2022

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Xaya or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors from legal and financial liability.

Oak Security GmbH