SOLIDIFIED

Summary

Audit Report prepared by Solidified covering the Wighawag universal forwarder smart contracts.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on 5 July 2021.

Audited Files

The source code has been supplied in the form of specific commits in a GitHub repository:

https://github.com/wighawag/universal-forwarder/tree/0ac0b2ece2feaee7ce0e5401480eca4016835b9c

Scope limited to the following files:

```
src/
├── ForwarderRegistry.sol
├── Test
│   ├── TestSpecificForwarderReceiver.sol
│   └── TestUniversalForwardingReceiver.sol
├── UniversalForwarder.sol
└── solc_0.7
    └── ERC2771
        ├── IERC2771.sol
        ├── IForwarderRegistry.sol
        ├── UsingAppendedCallData.sol
        ├── UsingSpecificForwarder.sol
        └── UsingUniversalForwarding.sol
```

Intended Behavior

The smart contracts implement a universal meta-transaction forwarder.

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 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	Medium	-
Code readability and clarity	Medium	-
Level of Documentation	Low	-
Test Coverage	High	-

Issues Found

Solidified found that the Wighawag contracts contain no security issues.

Disclaimer