Metal Token Audit

OPENZEPPELIN SECURITY | JUNE 30, 2017

Security Audits

The Metal team asked us to review and audit their new Metal Token contract code. We looked at their contracts and now publish our results.

The audited contracts can be found in their metal-token repo. The version used for this report is commit `d0ca13778c7c3ccc19d5fb2cb71c80588324bacf`.

Good work writing very minimal code and reusing existing contracts.

Here’s our assessment and recommendations, in order of importance.

Severe

No severe issues were found.

Warnings

OpenZeppelin vendoring

All contracts except `MetalToken` are from version 1.0.5 of OpenZeppelin. Consider installing the contracts from NPM instead of vendoring (copy-pasting) them into the repository.

Notes and Additional Information

- Good job using OpenZeppelin!

some discussion we agreed that this problem should be fixed elsewhere, and removed it from OpenZeppelin. It doesn’t seem to cause any issues in the Metal Token code, but consider removing it to reduce attack surface.
- The state variables `name`, `symbol`, `decimals` and `INITIAL_SUPPLY` should all be constants.
- `INITIAL_SUPPLY` should be defined using the `decimals` state variable as `66588888 * 10 ** decimals`. This is clearer and more future proof.
- `INITIAL_SUPPLY` amount is correct for the defined Metal token decimals (8).

Conclusions

No severe security issues were found. Some small changes were proposed to follow best practices and reduce potential attack surface.

Good work writing very minimal code and reusing existing contract modules.

Note that as of the date of publishing, the above review reflects the current understanding of known security patterns as they relate to the Metal Token contract. We have not reviewed the related Metal project. The above should not be construed as investment advice or an offering of tokens. For general information about smart contract security, check out our thoughts here.

Related Posts

intermediary designed to execute users' orders through routes...

Security Audits

OpenBrush is an open-source smart contract library written in the Rust programming language and the...

Security Audits

Ethereum. It is designed to be EVM-compatible and aims to...

Security Audits

OpenZeppelin

Defender Platform

Secure Code & Audit
Secure Deploy
Threat Monitoring
Incident Response
Operation and Automation

Services

Smart Contract Security Audit
Incident Response
Zero Knowledge Proof Practice

Learn

Docs
Ethernaut CTF
Blog

Company

About us
Jobs
Blog

Contracts Library

Docs

Privacy | Terms of Use