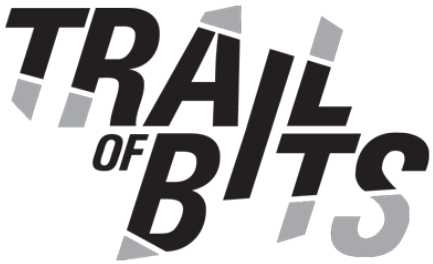

Golem Network Token Migration

Security Review Executive Summary

March 31, 2020

Prepared For:
Piotr Janiuk | *Golem Network*
viggith@golem.network

Prepared By:
Sam Moelius | *Trail of Bits*
sam.moelius@trailofbits.com

Josselin Feist | *Trail of Bits*
josselin@trailofbits.com

Michael Colburn | *Trail of Bits*
michael.colburn@trailofbits.com

Executive Summary

From March 23 through March 27, 2020, Golem Network engaged with Trail of Bits to review the security of the Golem Network Token migration. Trail of Bits conducted this assessment over the course of two person-weeks with three engineers working from commit `1fb991c87b2ddc0f0c76585e77e948de4cabeade` of the `gnt2` repository.

Trail of Bits performed this assessment through a combination of manual review, static analysis using Slither, and symbolic execution using Manticore. Particular attention was paid to situations in which the code could be misused, possibly accidentally. For those situations identified, we asked: are there ways to avoid them, or to ensure they can be recovered from?

Our efforts led to a total of eight findings ranging from high- to informational-severity. One high severity finding concerns a way in which the deployment could be backdoored. A second high severity finding concerns the potential for a certain type of phishing attack. Two medium-severity findings concern a type of transaction reordering attack and the lack of `chainID` verification by the `permit` function. Finally, four informational-severity findings concern the potential for an additional type of phishing attack, a race condition involving `permit` nonces, the fact that `migrateFrom` ignores `target.mint`'s return value, and the fact that `permit` does not allow for partial allowances.

In addition to the above, we identified some ways in which Golem Network's next token migration will necessarily differ from the present one. One of these concerns the potential for a certain type of phishing attack, mentioned above. A second is discussed in our Manticore results.

Finally, we identified two code quality issues.

Overall, the code is concise and neatly formatted. The use of well-tested components from existing codebases, namely MakerDAO's Multi-collateral DAI and OpenZeppelin's ERC20 implementation, further lends to its security. Trail of Bits recommends fixing the reported issues and carefully monitoring the migration.