Golem Network Token (GNT) Audit

NOVEMBER 8, 2016 | IN SECURITY AUDITS | BY OPENZEPPELIN SECURITY

We've been asked by the Golem team to review and audit their new token code. We looked at their contracts and now publish our results.

The audited contracts are at their golem-crowdfunding GitHub repo. The version used for this report is commit 50100b27a7c6841ed430a028d100f5d45ba08fb1. The main contract file is Token.sol.

Here's our assessment and recommendations, in order of importance:

Severe

We have not found any severe security problems with the code.

Potential problems

Timestamp usage

There's a problem with using timestamps and `now` (alias for `block.timestamp`) for contract logic, based on the fact that miners can perform some manipulation. In general, it's better not to rely on timestamps for contract logic. The solutions is to use `block.number` instead, and approximate dates with expected block heights and time periods with expected block amounts.

The GNTAllocation contract uses timestamps at several points. The risk of miner manipulation, though, is really low. The potential damage is also limited: miners could only slightly manipulate the developer lock period duration. We recommend the team to consider the potential risk and switch to `block.number` if necessary.

For more info on this topic, see this stack exchange question.

`block.number` is correctly used in the Token.sol file.

Use safe math

There are many unchecked math operations in the code. We couldn't find any related attack vectors on the GNT contract, but it's always better to be safe and perform checked operations. Consider using a safe math library, or performing pre-condition checks on any math operation.

The fact that GNT supply is limited to 820,000 ether (and thus at most 820,000,000 GNT assuming all tokens are created at the best possible price) helps prevent possible overflows.

Use of send

Use of `send` is always risky and should be analyzed in detail. Two occurrences found in line 185 of Token.sol and line 214 of Token.sol.

– Always check send return value: OK.
– Consider calling send at the end of the function: Risk is low because in one case, the contract `send` s to golemFactory, which is a trusted contract, and almost at the end of the function. Consider refactoring the code to have `send` at the end, though.
– Favor pull payments over push payments: No problems with push payment used, because `golemFactory` will be controlled by the Golem team. Bear in mind that if `send` at line 185 fails for any reason, the whole `finalize` call will fail.

Warnings

Remove duplicate code

Duplicate code makes it harder to understand the code's intention and thus, auditing the code correctly. It also increases the risk of introducing hidden bugs when modifying one of the copies of some code and not the others. We recommend the following to remove duplicate code:

- https://github.com/imapp-pl/golem-crowdfunding/blob/50100b27a7c6841ed430a028d100f5d45ba08fb1/contracts/Token.sol#L194-L196 are very similar to https://github.com/imapp-pl/golem-crowdfunding/blob/50100b27a7c6841ed430a028d100f5d45ba08fb1/contracts/Token.sol#L158-L165 and could be refactored into a single `createTokens` function.
- Lots of lines doing checks (e.g. if the contract is in funding mode) can be extracted as function modifiers to reduce function code length.

Bug Bounty

Formal security audits are not enough to be safe. We recommend implementing an automated contract-based bug bounty and setting a period of time where

security researchers from around the globe can try to break the token's invariants. For more info on how to implement automated bug bounties with OpenZeppelin, see this guide.

Additional Information and Notes

- Developer allocations should of course be replaced to actual addresses.
- Developer allocation percentages correctly add up to 100.00%.
- Consider separating the code that only works during funding phase from the rest of the code. This could make understanding how the token works better. See an example here (untested and not final code, just to show the idea)
- The comment at https://github.com/imapp-pl/golem-crowdfunding/blob/50100b27a7c6841ed430a028d100f5d45ba08fb1/contracts/Token.sol#L170 doesn't close the parenthesis. Consider changing it to "create GNT for the golemFactory (representing the company and developers)" and removing next line.
- Consider using a standard ERC20 token implementation. See here for a ready-to-use StandardToken.sol contract implementation.
- GolemNetworkToken constructor should probably check for required parameters. Consider adding checks for existence of all parameters.
- Owner of the `migrationAgent` address has a lot of power. Private key for that address should be stored with high security standards.

Conclusions

No severe security issues were found. Some changes were recommended to follow best practices and reduce potential attack surface. Overall code quality is very good! Functions are short, and all start with well-defined and well-ordered

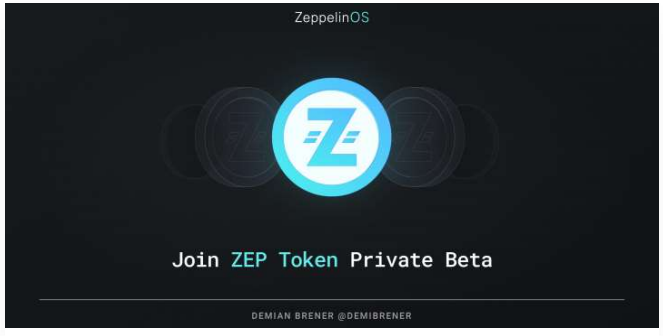
preconditions. In general, external effects are left to the end of functions. Most of the code is very well commented. 👍

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

Security Audits

- If you are interested in smart contract security, you can continue the discussion in our forum, or even better, join the team 🚀
- If you are building a project of your own and would like to request a security audit, please do so here.

RELATED POSTS



ANNOUNCEMENTS PRODUCT RELEASES

Announcing the ZEP Token Private Beta

Today we are excited to announce the ZEP Token Private Beta for ZeppelinOS — a network that uses...

READ MORE

ANNOUNCEMENTS PRODUCT RELEASES

How the ZEP Token Powers ZeppelinOS

ZeppelinOS is an operating system designed specifically for smart contracts. It's an open-source,...

READ MORE

PERSPECTIVES

On Tokens and Crowdsales

Companies we've never heard of are IPO'ing before our very eyes. Instead of ringing the IPO bell...

READ MORE

Email*

Get our monthly news roundup

由 reCAPTCHA 提供保护
隐私权 - 使用条款

SIGN UP

Products
- Contracts
- Defender

Security
- Security Audits

Learn
- Docs
- Forum
- Ethernaut

Company
- Website
- About
- Jobs
- Logo Kit

Privacy | Terms of Service