Serpent Compiler Audit

OPENZEPPELIN SECURITY | JULY 28, 2017

Security Audits

The Augur team asked us to review and audit the Serpent compiler. We looked at the code and now publish our results.

The audited project can be found in the ethereum/serpent GitHub repository. The version used for this report is commit `ad53fa2a8a496448d58ef9137959b4a1e86b14d7`.

We have found the Serpent project to be of very low quality. It is untested, there's very little documentation, and language design is very flawed. Serpent should not be considered safe to use unless many critical problems are fixed.

The full report can be found here, and a list summary of the issues ordered by severity can be found next.

Critical Severity

- Very low quality tool and language
- Development stalled
- No tests
- Language is untyped
- Invalid syntax is accepted by the parser and compiler
- Compiler does not fail on non-initialized variables
- Can overwrite local memory location when accessing array out of bounds
- Can overwrite storage location when accessing array out of bounds

- No internal functions
- Difference between short and long strings
- Python reserved keywords remain unimplemented and are interpreted as uninitialized variables
- Special any functions are chained, while shared are overwritten
- Unexpected ABI generated from syntax error in method signature
- Inline return statements are ignored
- Return is not a reserved keyword
- Medium sized strings are parsed as variables
- Can overwrite local memory location when using setch with out of bounds index

Medium severity

- Lack of documentation
- Consider using Viper
- Broken and confusing examples
- Serpent's sha3 function is not the standard SHA-3
- Integer operations are signed
- + operator on short strings does not concatenate
- Bad send API design
- Variable names can begin with digits
- Python boolean constants are unimplemented
- Serpent's send behavior is different to Solidity's send
- Constructor cannot accept parameters
- No way to declare minimum compiler version
- No error reporting on non-existent commands
- Non-standard ABI names generated from CLI
- Parser internal tokens clash with user-defined ones
- Can overwrite non initialized variables
- Can declare arrays of negative size
- Array sizes can overflow
- Compiler gives no warning when accessing uninitialized arrays

- Several documentation errors
- Use of magic constants
- Code mixes tabs and spaces
- Inconsistent casing
- Bad CLI parameter parsing
- Bad CLI error handling
- Makefile does not specify dependencies correctly
- Length function does not check argument types
- Local variables defined in shared or any functions leak into user functions
- Inconsistent usage of outitems vs outsz
- Array arguments should be a pointer to first element

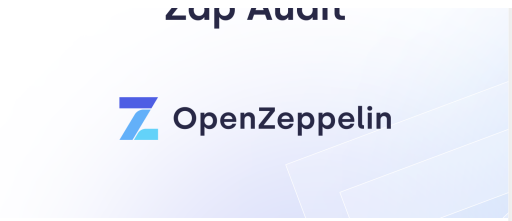
Conclusions

Eight critical security issues were found and explained, along with recommendations on how to fix them. Some additional changes were proposed to follow best practices and reduce potential attack surface.

We have found the Serpent project to be of very low quality. It is untested, there's very little documentation, and language design is very flawed. Serpent should not be considered safe to use unless many critical problems are fixed.

Note that as of the date of publishing, the above review reflects the current understanding of known security patterns as they relate to the Serpent compiler. We have not reviewed the related Augur 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



Beefy Zap Audit

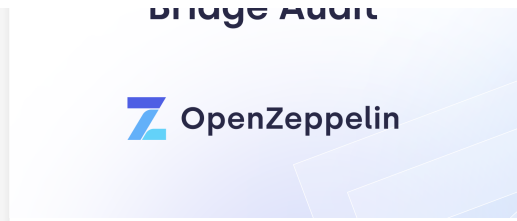
BeefyZapRouter serves as a versatile intermediary designed to execute users' orders through routes...

Security Audits

OpenBrush Contracts Library Security Review

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

Security Audits



Linea Bridge Audit

Linea is a ZK-rollup deployed on top of Ethereum. It is designed to be EVM-compatible and aims to...

Security Audits