April 2026

SBOMAUDIT

A FRAMEWORK FOR VERIFICATION OF SBOMS USING BINARY ANALYSIS

Oscar Bergström
osbergs@kth.se

Examiner:
Martin Monperrus

Supervisor:
Eric Cornelissen

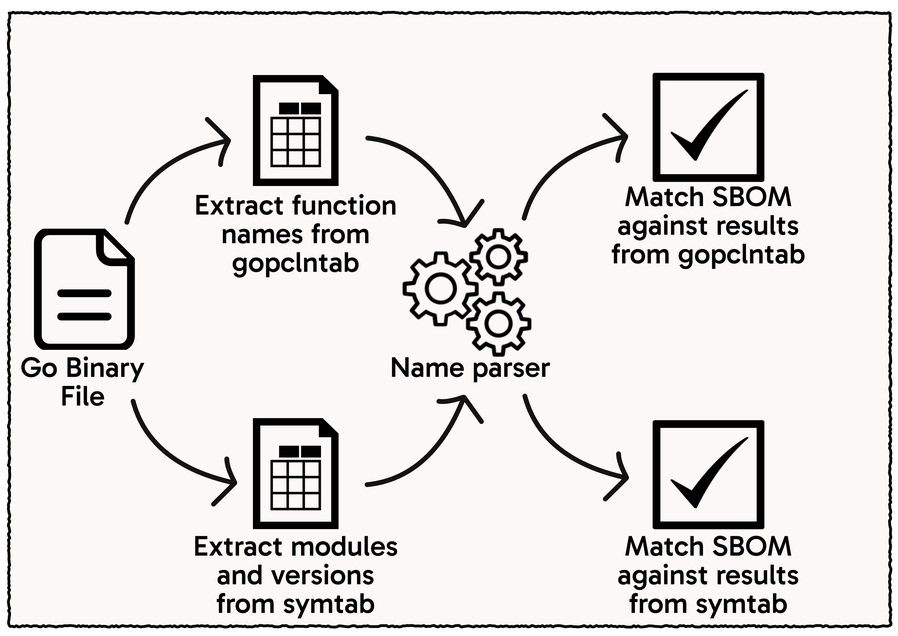
The Problem

SBOMs are meant to show what software is made of, but today they are often trusted without being verified. This creates a risk, since the listed dependencies may not actually match what is in the compiled program. This study addresses that by exploring how SBOMs can be checked against stripped & non-stripped Go binaries without revealing proprietary information.

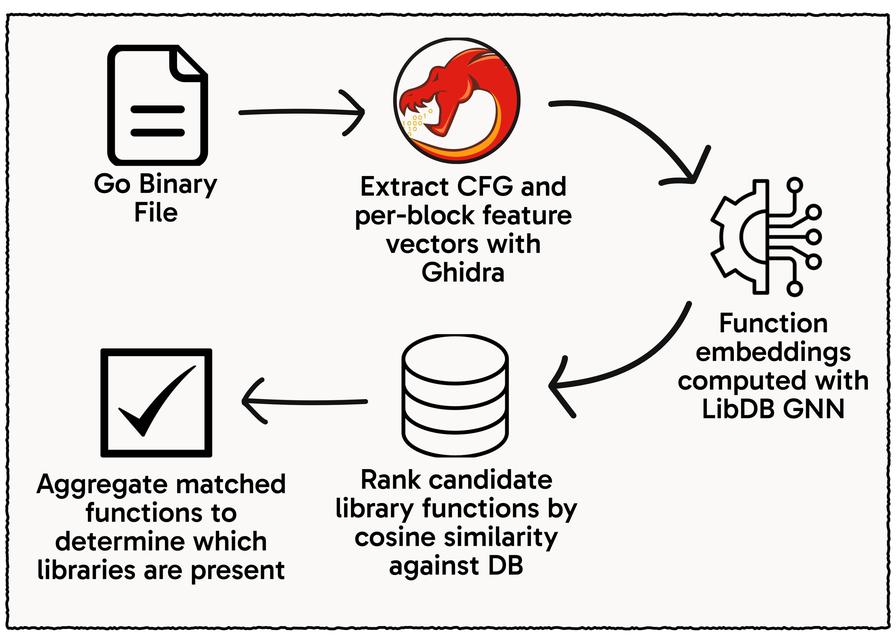
Methodology

- Utilize LibDB [1] in combination with Ghidra [2] for library fingerprinting to identify embedded or linked libraries in the binary.
- Extract information from the Go Program Counter Line Table [3] and parse the output.
- Examine the symbol table [3] to extract dependency and package information.
- Evaluate discrepancies to determine the accuracy and completeness of the SBOM

symtab & gopclntab Examination



Library Fingerprinting Pipeline



Preliminary Results

- Binary analysis looks to be a promising solution for verifying SBOMs in Go.
- Symbol table examination is accurate when the binary is not stripped. There are challenges arising from differences in formatting between SBOM data and symbol table output.
- Go Program Counter Line Table examination is providing high accuracy for both stripped and unstripped binaries.
- Transitive dependency data is publicly & easily available for all Go modules. Making dependency mapping trivial.
- Fingerprinting with LibDB is proving difficult with a limited amount of Go modules having unique functions for identification. Accuracy is high when identifiers are available.

References

[1] W. Tang, et.al "LibDB:an effective and efficient framework for detecting third-party libraries in binaries,"

[2] NSA - https://github.com/nationalsecurityagency/ghidra

[3] Google - https://cloud.google.com/blog/topics/threat-intelligence/golang-internals-symbol-recovery/