Go Threat Model

- Threat Model
  - Build Safety
  - Runtime Execution
  - Memory Safety
  - Trust Boundaries
  - Environment Control
- Packages With Their Own Models

This document defines the general threat model for the Go toolchain and standard library. In the absence of package documentation defining a threat model, the model described in this document should be assumed to apply to all packages in the standard library.

Threat Model

Build Safety

Building Go code is assumed to be safe and should have no side-effects, such as unexpected execution.

Runtime Execution

We generally do not consider the execution of malicious code to be a relevant security issue. A user who is familiar with Go is assumed to understand what they are executing.

Memory Safety

It is assumed that in the absence of usages of the `unsafe` package, memory safety is guaranteed by the runtime.

Trust Boundaries

APIs which are reasonably expected to accept arbitrary user provided input are assumed to be hardened against panics and arbitrary resource consumption.

Passing garbage to an API resulting in unexpected output is not considered a security issue.

Environment Control

It is assumed that the local system is safe. Attacks which rely on the OS already being compromised are not considered relevant. For instance we do not consider attacker control over the filesystem, environment variables, such as PATH, or memory access or control to be part of our model.

Packages With Their Own Models

- encoding/json
- encoding/json/v2
- encoding/json/jsontext
- encoding/gob
- html/template
- image
- debug/pe
- debug/macho
- debug/dwarf
- debug/plan9obj
- debug/elf

□□□ #2
□ □□□ □□□ 2026-10-26 20:26:29 UTC
□ □□□ □□□ 2026-10-26 20:26:29 UTC