Groups Conversations Search conversations within golang-a...

golang-announce

Conversations

About

[security] Go 1.26.3 and Go 1.25.10 are released

384 views

anno...@golang.org
to golan...@googlegroups.com

Hello gophers,

We have just released Go versions 1.26.3 and 1.25.10, minor point releases.

These releases include 11 security fixes following the security policy:

- cmd/go: malicious module proxy can bypass checksum database

  A malicious module proxy could exploit a flaw in the go command's validation of module checksums to bypass checksum database validation.

  This vulnerability affects any user using an untrusted module proxy (GOMODPROXY) or checksum database (GOSUMDB).

  A malicious module proxy can serve altered versions of the Go toolchain. When selecting a different version of the Go toolchain than the currently installed toolchain (due to the GOTOOLCHAIN environment variable, or a go.work or go.mod with a toolchain line), the go command will download and execute a toolchain provided by the module proxy. A malicious module proxy can bypass checksum database validation for this downloaded toolchain.

  Since this vulnerability affects the security of toolchain downloads, setting GOTOOLCHAIN to a fixed version is not sufficient. You must upgrade your base Go toolchain.

  The go tool always validates the hash of a toolchain before executing it, so fixed versions will refuse to execute any cached, altered versions of the toolchain.

  The go tool trusts go.sum files to contain accurate hashes of the current module's dependencies. A malicious proxy exploiting this vulnerability to serve an altered module will have caused an incorrect hash to be recorded in the go.sum. Users who have configured a non-trusted GOPROXY can determine if they have been affected by running "rm go.sum ; go mod tidy ; go mod verify", which will revalidate all dependencies of the current module.

  The specific flaw in more detail:

  The go command consults the checksum database to validate downloaded modules, when a module is not listed in the go.sum file. It verifies that the module hash reported by the checksum database matches the hash of the downloaded module. If, however, the checksum database returns a successful response that contains no entry for the module, the go command incorrectly permitted validation to succeed.

  A module proxy may mirror or proxy the checksum database, in which case the go command will not connect to the checksum database directly. Checksums reported by the checksum database are cryptographically signed, so a malicious proxy cannot alter the reported checksum for a module. However, a proxy which returns an empty checksum response, or a checksum response for an unrelated module, could cause the go command to proceed as if a downloaded module has been validated.

  The go command now properly checks checksum database responses to ensure that the expected module signature is present, not just that if a signature is present it matches the expectation.

  Thanks to Mundur (https://github.com/M0nd0R) for reporting this issue.

  This is CVE-2026-42501 and Go issue https://go.dev/issue/79070.

- net/http/httputil: ReverseProxy forwards queries with more than urlmaxqueryparams parameters

  When used with a Rewrite function, or a Director function which parses query parameters, ReverseProxy sanitizes the forwarded request to remove query parameters which are not parsed by url.ParseQuery. ReverseProxy did not take ParseQuery's limit on the total number of query parameters (controlled by GODEBUG=urlmaxqueryparams=N) into account. This could permit ReverseProxy to forward a request containing a query parameter that was not visible to the Rewrite function.

  For example, the query "a1=x&a2=x&...&a10000=x&hidden=y" could forward the parameter "hidden=y" while hiding it from the proxy's Rewrite function.

  ReverseProxy now avoids forwarding parameters that exceed the ParseQuery limit.

  This is CVE-2026-39825 and Go issue https://go.dev/issue/78948.

- net: panic in Dial and LookupPort when handling NUL byte on Windows

Privacy • Terms

This is CVE-2026-39836 and Go issue https://go.dev/issue/79006.

- net/mail: quadratic string concatenation in consumePhrase

  Pathological inputs could cause DoS through consumePhrase when parsing an email address according to RFC 5322.

  This is CVE-2026-42499 and Go issue https://go.dev/issue/78987.

- net/mail: quadratic string concatentation in consumeComment

  Well-crafted inputs reaching ParseAddress, ParseAddressList, and ParseDate were able to trigger excessive CPU exhaustion and memory allocations.

  This is CVE-2026-39820 and Go issue https://go.dev/issue/78566.

- cmd/go: "go bug" follows symlinks in predictable temporary filenames

  The "go bug" command wrote to two files with predictable names in the system temporary directory (for example, "/tmp").

  An attacker with access to the temporary directory could create a symlink in one of these names, causing "go bug" to overwrite the target of the symlink.