Go mod’s lesser known features

for software supply chain security

Dr. Tony Worm
verdverm.com

Go Modules

A **module** is a collection of packages that are released, versioned, and distributed together

A **package** is a collection of source files in the same directory that are compiled together

The go.mod file

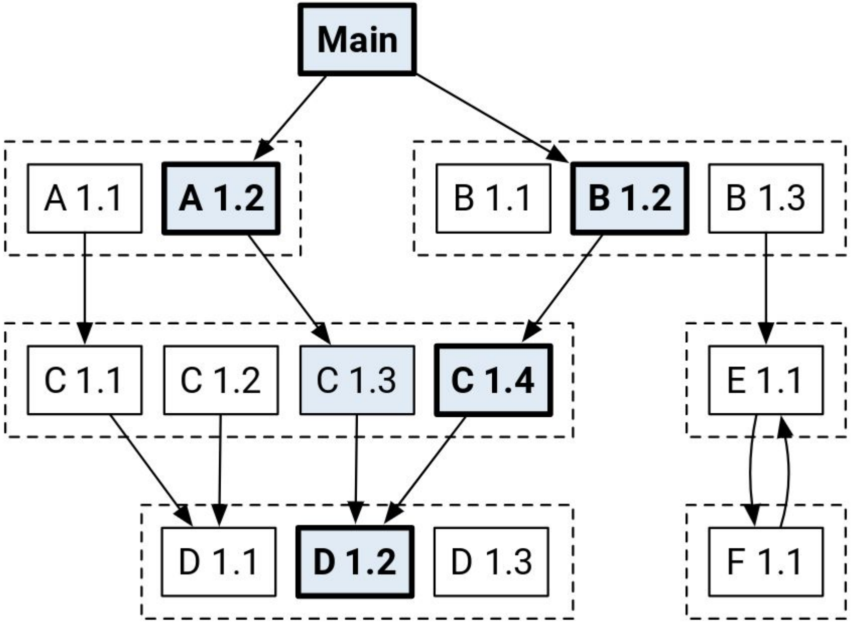
```
// module name
module github.com/org/module


// module dependencies
require (
        github.com/foo/bar  v0.1.2
        github.com/cow/moo  v1.2.3
        mydomain.com/gopher v0.2.3-beta1
)
```

Minimum Version Selection (MVS)

Deterministic and reproducible algorithm for dependency selection without a lockfile.

No SAT solver, avoids NP-completeness



Directives in go.mod

```
module example.com/my/thing


go 1.16


require example.com/other/thing v1.0.2

require example.com/new/thing/v2 v2.3.4

exclude example.com/old/thing v1.2.3

replace example.com/bad/thing v1.4.5 => example.com/good/thing v1.4.5

retract [v1.9.0, v1.9.5]
```

Environment Variables

GOMODCACHE	directory for module related files
GOPRIVATE	module globs to handle as private
GOPROXY	ordered list of module proxies to use
GONOPROXY	module globs to fetch directly
GOSUMDB	ordered list of sumdb hosts to use
GONOSUMDB	module globs to omit remote sumdb checks on
GOVCS	sets VCS tools allowed for public and private access
GOINSECURE	globs to allow fallback to http on

Hashes and go.sum

Go computes a cryptographic hash on module download

Stores in a **go.sum** file for each dependency

Compares against known values in **go.sum** and beyond

Local module cache

Go maintains a shared module cache on your local system

Read-only source + module hash

Also has reviously built artifacts

Global module cache and sumdb

Go team maintains global proxies modules and hashes

sumdb is global merkel-tree for module hashes powered by the Google/Trillian project

They take privacy seriously as evident in their conversation on GitHub issues

Module naming

Domain must be the first part	Code resolved to origin
Only ascii, digits, and limited punctuation	Homoglyph attacks
Cannot begin or end with slash or dot	Relative pathing attacks

Only secure remotes

Go will only talk to **https** and **git+ssh** by default

GOVCS enables more tools and protocols

GOINSECURE enables insecure protocols

Private module support

Fetch modules from private code repositories

Prevent private modules from being publicly indexed

Run private proxies and sumdb

Authentication is through VCS tool config or .netrc

Prevent Dependency Confusion

Dependency confusion is when a public package with the same name as an internal package is fetched

Requiring domains to start module and import paths

Ignoring replace directives in dependencies

Malicious version changes

Replacing a tag - practically impossible due to the global proxies

Creating a new tag - Go only selects from listed versions, no ranges

No pre or post hooks

Go lacks any pre or post hooks for fetch, build, or install.

Rules out a class of attacks, such as those seen with NPM.

Information in the binary

Go adds the dependency information into the binary.

```
go version -m $(which binary)
```

Go 1.18 will include build flags, environment variables, and VCS details

Reproducible builds

Go aims for 100% reproducible builds

MVS & dependency management is key to this

Works even when cross-compiling

Can work with CGO

Learn more

Module Reference

Go & Versioning

Original Proposal

GitHub Issues

Slides and post available at verdverm.com/go-mods

Questions?