GoTalks 30.01.2024

Meetup news

Breaking change 😳

Docker images

- https://github.com/docker-library/golang/issues/502
    - existing tag lost support for platform
    - `golang:1.21.5 on linux/arm/v6 : no match for platform in manifest`
- not the first time that issues like this happened with docker build team 😳

Meetup news

Go 1.22

- expected to be released in February 2024
- go1.22rc2 available

Accepted

- all: end support for macOS 10.15 in Go 1.23 #64207
- x/crypto/ssh: support server-side Diffie-Hellman Group Exchange #54743
- x/tools/cmd/deadcode: add new command #63501

Go tools

- govulncheck
- go-licenses
- task

Govulncheck

- 13 July 2023
    - Govulncheck v1.0.0
- Go’s support for vulnerability management
- Go 1.18 or later

Govulncheck

Instalation

```
go install golang.org/x/vuln/cmd/govulncheck@latest
```

Usage

```
govulncheck ./...
```

Govulncheck CI

```
time=2024-01-10T16:27:17.968Z level=INFO source=main.go:57 msg="Starting GoVulnCheck
time=2024-01-10T16:27:17.968Z level=INFO source=main.go:73 msg="Current branch: vX.Y
Scanning your code and 1139 packages across 234 dependent modules for known vulnerab
Vulnerability #1: GO-2023-2402
    Man-in-the-middle attacker can compromise integrity of secure channel in
    golang.org/x/crypto
  More info: https://pkg.go.dev/vuln/GO-2023-2402
  Module: golang.org/x/crypto
    Found in: golang.org/x/crypto@v0.14.0
    Fixed in: golang.org/x/crypto@v0.17.0
    Example traces found:
      #1: path/to/mypackage/src.go:132:24:
          mypackage.Function calls bytes.Buffer.ReadFrom,
          which calls ssh.extChannel.Read
Your code is affected by 1 vulnerability from 1 module.
Share feedback at https://go.dev/s/govulncheck-feedback.
ERROR: Job failed: exit code 1
```

Govulncheck - Actions

- Option 1: Upgrade to a fixed version
- Option 2: Stop using the vulnerable symbol(s)

```
module github.com/meetup/example-bad

go 1.21.0

require golang.org/x/net v0.14.0
```

Govulncheck - Actions

- Option 1: Upgrade to a fixed version
- Option 2: Stop using the vulnerable symbol(s)

```
module github.com/meetup/example-good

go 1.21.0

require golang.org/x/net v0.20.0
```

Govulncheck - Problems

- govulncheck is scanning everything!

```
=== Informational ===
Vulnerability #1: GO-2023-2163
    Curve KeyPairs fail to encrypt in github.com/nats-io/nkeys
  More info: https://pkg.go.dev/vuln/GO-2023-2163
  Module: github.com/nats-io/nkeys
    Found in: github.com/nats-io/nkeys@v0.4.4
    Fixed in: github.com/nats-io/nkeys@v0.4.6
```

- not in `go.mod` file !!

Govulncheck - Problems

similar to what is used for tools

'solution'

```go
package deps

import (
        _ "github.com/nats-io/nkeys"
)
```

Govulncheck - Problems

```
govulncheck ./...
Scanning your code and 1512 packages across 266 dependent modules
for known vulnerabilities...

Killed
```

- github.com/golang/go/issues/56391
  - Closing for now as "working as intended."

go-licenses

A lightweight tool to report on the licenses used by a Go package and its dependencies

Installation

```
go install github.com/google/go-licenses@latest
```

go-licenses

- Report

```
go-licenses report <package> [package]
go-licenses report <package> [package] --template=<template_file>
```

- Save

```
go-licenses save <package> [package] --save_path=<save_path>
```

- Check

```
go-licenses check <package> [package]
```

go-licenses - usage

- local

```
go-licenses report ./cmd/my-app
```

- remote

```
go-licenses report github.com/google/go-licenses
```

go-licenses - usage

- go-licenses writes the report to stdout and info/warnings/errors logs to stderr.
- csv format
  - package
  - url
  - license type

```
go-licenses report github.com/google/go-licenses > licenses.csv
```

```
go-licenses report github.com/google/go-licenses \
  > licenses.csv 2> errors
```

go-licenses - usage

```
go-licenses save "github.com/google/go-licenses" \
  --save_path="/tmp/go-licenses-cli"
```

This command analyzes a binary/package's dependencies and determines what needs to be redistributed alongside that binary/package in order to comply with the license terms. This typically includes the license itself and a copyright notice, but may also include the dependency's source code. All of the required artifacts will be saved in the directory indicated by --save_path.

go-licenses - CI - setup

- Prepare file with licenses

```
$ cat .allowed_license_types
MIT,BSD-2-Clause,BSD-3-Clause,Apache-2.0,ISC
```

- could be ENV variable

go-licenses - CI

- simple

```
go-licenses check --allowed_licenses="${ALLOWED_LICENSES}" \
  ./cmd/my-app
```

- exclude current project

```
PROJECT=$(shell go list -m)
go-licenses check --allowed_licenses="${ALLOWED_LICENSES}" \
  --ignore "${PROJECT}" ./cmd/my-app
```

go-licenses - CI

- more complex example

```
PROJECT=$(shell go list -m)
go-licenses check --allowed_licenses="${ALLOWED_LICENSES}" \
  --ignore "${PROJECT}" --ignore my.git.xyz/my/repo \
  ./cmd/my-app
```

go-licenses - scanning

- csv format
  - package
  - url
  - license type

```
W0114 00:16:14.026486 1046589 library.go:101] "golang.org/x/sys/unix" contains non-Go
  code that can't be inspected for further dependencies:
/....../pkg/mod/golang.org/x/sys@v0.15.0/unix/asm_linux_amd64.s
github.com/fsnotify/fsnotify,https://github.com/fsnotify/fsnotify/blob/v1.7.0/LICENSE,BSD-3-Clause
golang.org/x/sys/unix,https://cs.opensource.google/go/x/sys/+/v0.15.0:LICENSE,BSD-3-Clause
```

github.com/fsnotify/fsnotify

https://github.com/fsnotify/fsnotify/blob/v1.7.0/LICENSE

BSD-3-Clause

Task

taskfile.dev

Task is a task runner / build tool that aims to be simpler and easier to use than, for example, GNU Make

```
version: '3'

tasks:
  hello:
    cmds:
      - echo 'Hello World from Task!'
    silent: true
```

Task - Installation

taskfile.dev/installation

Use Go 😃

```
go install github.com/go-task/task/v3/cmd/task@latest
```

Task - golangci-lint

```yaml
env:
  GOLANGCI_LINT_VERSION: 1.55.2

tasks:
  lint:
    status:
      - bin/golangci-lint --version | grep -q "{{.GOLANGCI_LINT_VERSION}}"
    cmds:
      - curl -sSfL https://raw.githubusercontent.com/golangci/golangci-lint/master/install.sh |
        sh -s -- -b {{.TASKFILE_DIR}}/bin "v{{.GOLANGCI_LINT_VERSION}}"
```

Task - golangci-lint

```
$ task lint
task: [lint-check-version] curl -sSfL https://raw.githubusercontent.com/g
golangci/golangci-lint info checking GitHub for tag 'v1.55.2'
golangci/golangci-lint info found version: 1.55.2 for v1.55.2/linux/amd64
golangci/golangci-lint info installed project/path/bin/golangci-lint
```

Task - gofumpt

```yaml
tasks:
  gofumpt:
    internal: true
    status:
      - which gofumpt
    cmds:
      - go install mvdan.cc/gofumpt@latest
```

- `internal: true`
  - internal tasks are not shown in the task list

Task - tools

```
tasks:
  tools:
    desc: 'are all tools are installed and have correct version'
    cmds:
      - task: gofumpt
      - task: lint-check-version

  gofumpt:
    internal: true
    status:
      - which gofumpt
  ...
  ...
```

Task - tools

```
$ task tools
task: Task "gofumpt" is up to date
task: Task "lint-check-version" is up to date
```

```
$ task tools
task: Task "gofumpt" is up to date
task: [lint-check-version] curl -sSfL https://raw.githubuserconte
golangci/golangci-lint info checking GitHub for tag 'v1.55.2'
golangci/golangci-lint info found version: 1.55.2 ...
golangci/golangci-lint info ...
```

Task - dependancies

```
tasks:
  lint:
    desc: 'run Go linter checks'
    deps: [lint-check-version]
    cmds:
      - bin/golangci-lint run --timeout 5m --color always --max-i
  format:
    desc: 'Format the code'
    deps: [gofumpt]
    cmds:
      - gofumpt -l -w .
```

Task - dependancies

```
deps: [gofumpt]
```

```
$ task format
task: Task "gofumpt" is up to date
task: [format] gofumpt -l -w .
```

Task - gotest

```
tasks:
  test:
    desc: 'run Go tests'
    cmds:
      - go test ./...
  e2e:
    ...
```

Task - advanced

Task - Go's template engine

- Go templates

```
vars:
  NUM: "{{ default \"1\" .NUM}}"
  GO_VERSION: 1.21
  DOCKER_VERSION: 24.0
tasks:
  print-date:
    cmds:
      - echo {{now | date "2006-01-02"}}
```

Task - Go's template engine

- OS: Returns the operating system.
    - Possible values are windows, linux, darwin (macOS) and freebsd.
- ARCH: return the architecture
    - Task was compiled to: 386, amd64, arm or s390x.
- splitLines, catLines, toSlash, fromSlash, exeExt
- shellQuote, splitArgs, joinPath, relPath, merge, spew

Task - .env

.env

```
KEYNAME=VALUE
```

```
env:
  ENV: testing
dotenv: ['.env', '{{.ENV}}/.env.', '{{.HOME}}/.env']

tasks:
  greet:
    cmds:
      - echo "Using $KEYNAME and endpoint $ENDPOINT"
```

Task - multiple taskfiles

```
version: '3'

includes:
  docs: ./doc # will look for ./doc/Taskfile.yml
  docker: ./DockerTasks.yml
```

Task - OS

```
version: '3'

includes:
  build: ./Taskfile_{{OS}}.yml
```

Task - looping

```
version: '3'

tasks:
  default:
    cmds:
      - for: ['foo.txt', 'bar.txt']
        cmd: cat {{ .ITEM }}
```

Task - looping

```
version: '3'

tasks:
  default:
    sources:
      - foo.txt
      - bar.txt
    cmds:
      - for: sources
        cmd: cat {{ .ITEM }}
```

Task - looping

```
version: '3'

tasks:
  default:
    vars:
      MY_VAR: foo.txt bar.txt
    cmds:
      - for: { var: MY_VAR }
        cmd: cat {{.ITEM}}
```

Task - advanced

```
go:
  cmds:
    - for: ['1.20',  '1.21']
      task: general-build
      vars:
        IMAGE: golang
        VERSION: '{{.ITEM}}'
```

Task - advanced

```
general-build:
  internal: true
  deps:
    - task: greeting
      vars: {DESC: '{{.DESC}}'}
  cmds:
    - docker build -f docker/{{.FOLDER}}/Dockerfile
      --build-arg GO_VERSION={{.GO_VERSION}}
      -t {{.REGISTRY}}/{{.OUT_IMAGE}}:{{.VERSION}}{{.POSTFIX}}
    - docker push {{.REGISTRY}}/{{.OUT_IMAGE}}:{{.VERSION}}{{.P
```

Task - other

- Optional includes
- Internal includes
- Vars of included Taskfiles
- Namespace aliases
- Task directory
- Platform specific tasks and commands
- By fingerprinting locally generated files and their sources
- Dry run mode

Task - defer

```
version: '3'

tasks:
  default:
    cmds:
      - mkdir -p tmpdir/
      - defer: rm -rf tmpdir/
      - echo 'Do work on tmpdir/'
```

Task - Warning Prompts

```
tasks:
  example:
    cmds:
      - task: not-dangerous
      - task: dangerous
  not-dangerous:
    cmds:
      - echo 'not dangerous command'
  dangerous:
    prompt: Dangerous command... Do you want to continue?
    cmds:
      - echo 'dangerous command'
```

Task - Watch tasks

```yaml
interval: 500ms

tasks:
  build:
    desc: Builds the Go application
    watch: true
    sources:
      - '**/*.go'
    cmds:
      - go build # ...
```