# What’s new in Go

Nils Dijk – June 2023

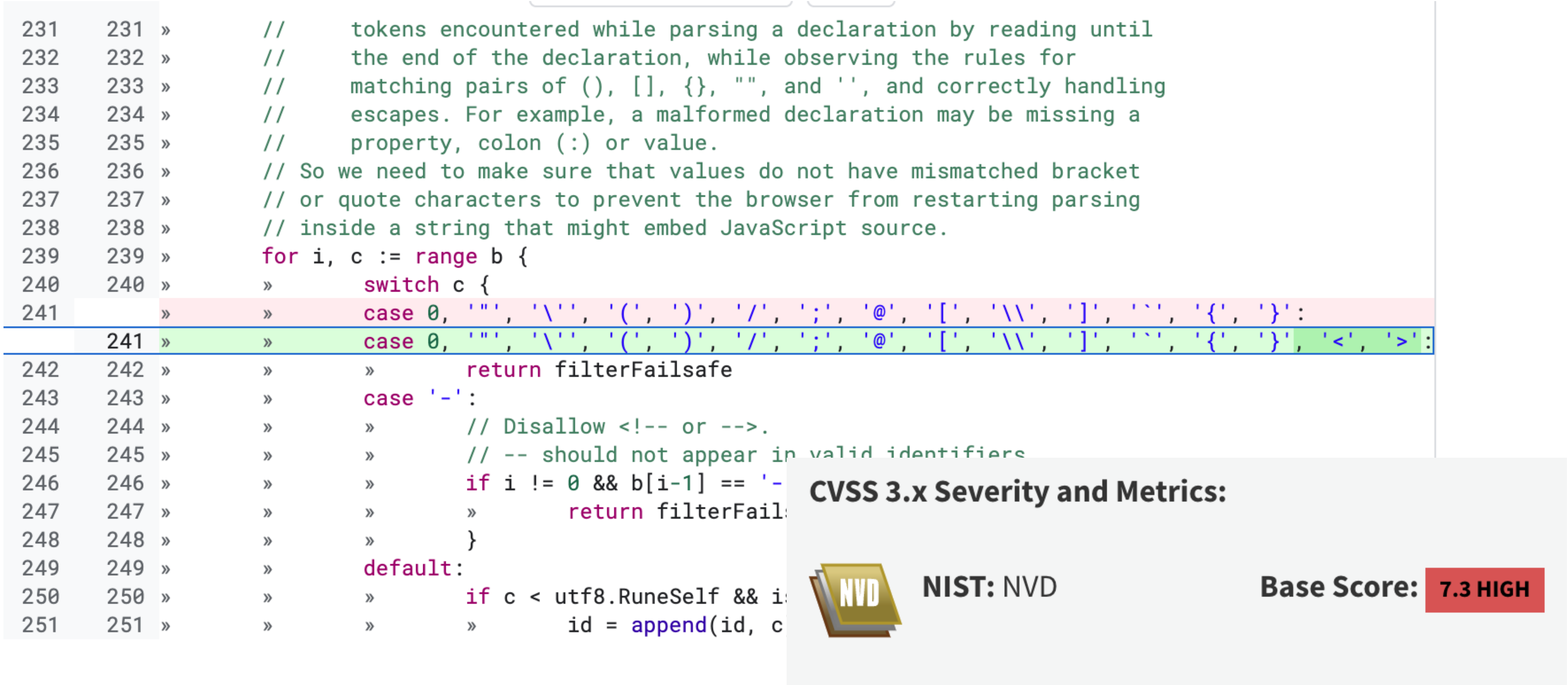
Go meetup @ Creative Fabrica – May 2nd

# 2 Security releases

- Go 1.20.4 / 1.19.9 – May 2nd
  - html/template: improper sanitization of CSS values
  - html/template: improper handling of JavaScript whitespace
  - html/template: improper handling of empty HTML attributes

- Go 1.20.5 / 1.19.10 – June 6th
  - cmd/go: cgo code injection
  - runtime: unexpected behavior of setuid/setgid binaries
  - cmd/go: improper sanitization of LDFLAGS

# html/template: improper sanitization of CSS values



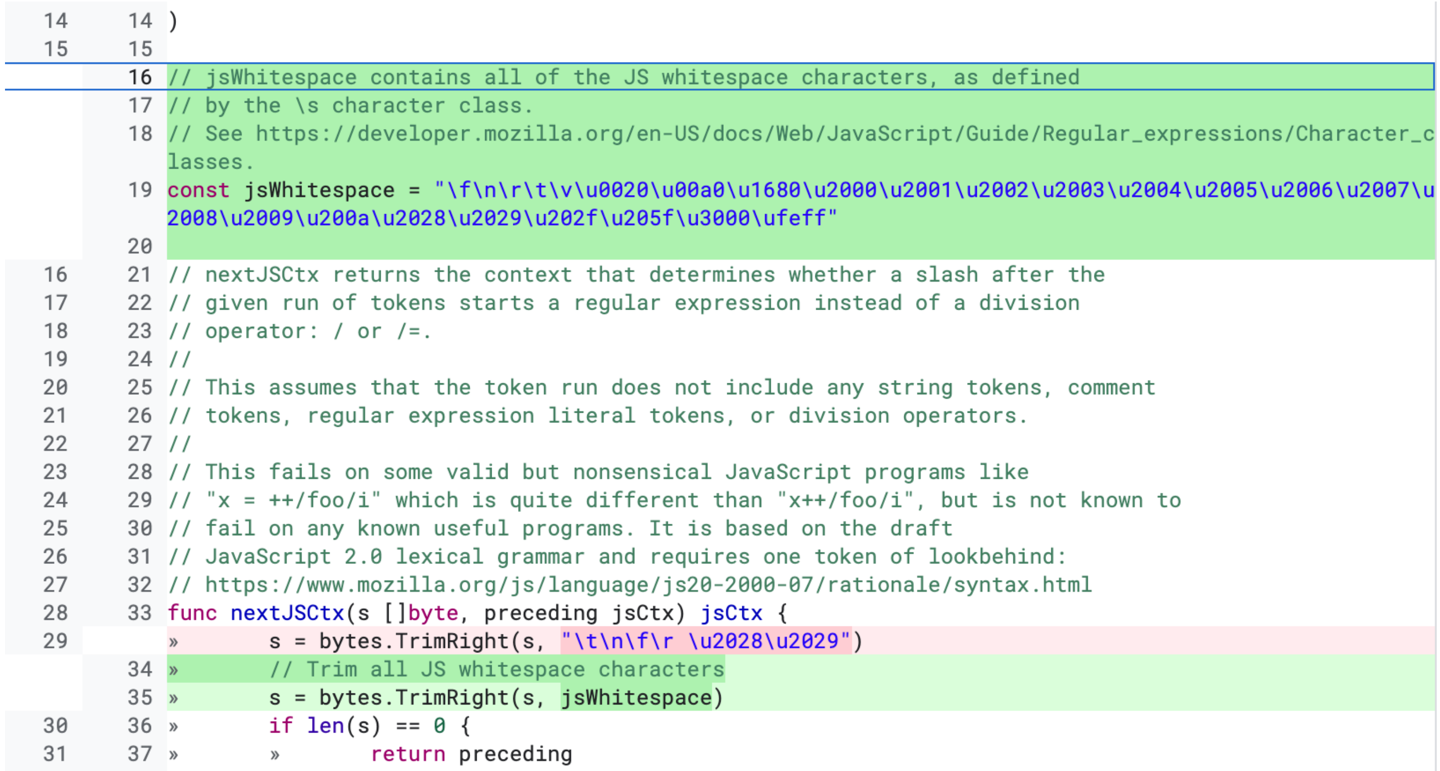
```
	//	tokens encountered while parsing a declaration by reading until
	//	the end of the declaration, while observing the rules for
	//	matching pairs of (), [], {}, "", and '', and correctly handling
	//	escapes. For example, a malformed declaration may be missing a
	//	property, colon (:) or value.
	// So we need to make sure that values do not have mismatched bracket
	// or quote characters to prevent the browser from restarting parsing
	// inside a string that might embed JavaScript source.
	for i, c := range b {
		switch c {
		case 0, '"', '\'', '(', ')', '/', ';', '@', '[', '\\', ']', '`', '{', '}':
		case 0, '"', '\'', '(', ')', '/', ';', '@', '[', '\\', ']', '`', '{', '}', '<', '>':
			return filterFailsafe
		case '-':
			// Disallow <!-- or -->.
			// -- should not appear in valid identifiers
			if i != 0 && b[i-1] == '-
				return filterFail
			}
		default:
			if c < utf8.RuneSelf && i
				id = append(id, c
```

**CVSS 3.x Severity and Metrics:**

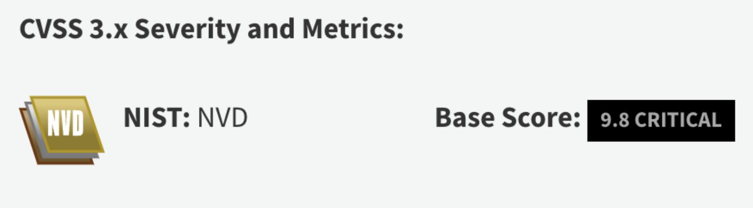
**NIST:** NVD

**Base Score:** 7.3 HIGH

# html/template: improper handling of JavaScript whitespace



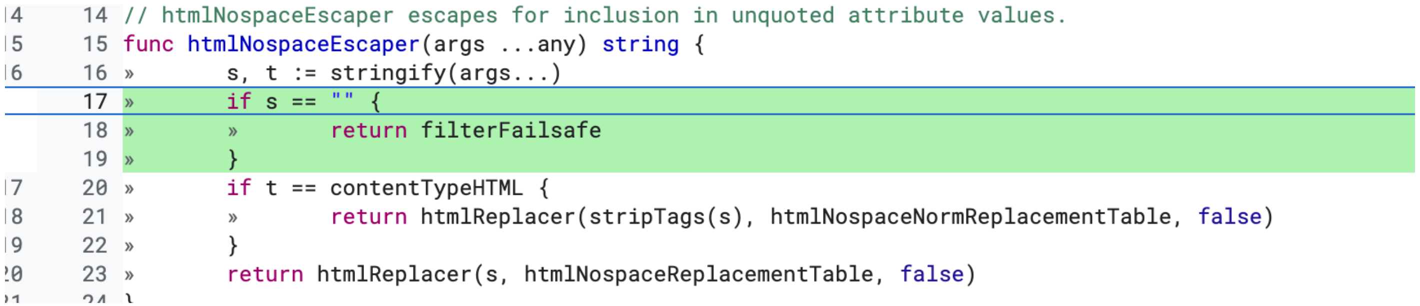
```
)

// jsWhitespace contains all of the JS whitespace characters, as defined
// by the \s character class.
// See https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions/Character_classes.
const jsWhitespace = "\f\n\r\t\v\u0020\u00a0\u1680\u2000\u2001\u2002\u2003\u2004\u2005\u2006\u2007\u2008\u2009\u200a\u2028\u2029\u202f\u205f\u3000\ufeff"

// nextJSCtx returns the context that determines whether a slash after the
// given run of tokens starts a regular expression instead of a division
// operator: / or /=.
//
// This assumes that the token run does not include any string tokens, comment
// tokens, regular expression literal tokens, or division operators.
//
// This fails on some valid but nonsensical JavaScript programs like
// "x = ++/foo/i" which is quite different than "x++/foo/i", but is not known to
// fail on any known useful programs. It is based on the draft
// JavaScript 2.0 lexical grammar and requires one token of lookbehind:
// https://www.mozilla.org/js/language/js20-2000-07/rationale/syntax.html
func nextJSCtx(s []byte, preceding jsCtx) jsCtx {
	s = bytes.TrimRight(s, "\t\n\f\r \u2028\u2029")
	// Trim all JS whitespace characters
	s = bytes.TrimRight(s, jsWhitespace)
	if len(s) == 0 {
		return preceding
	}

	// All cases below are in the single-byte UTF-8 group.
	switch c, n := s[len(s)-1], len(s); c {
	case '+', '-':
```



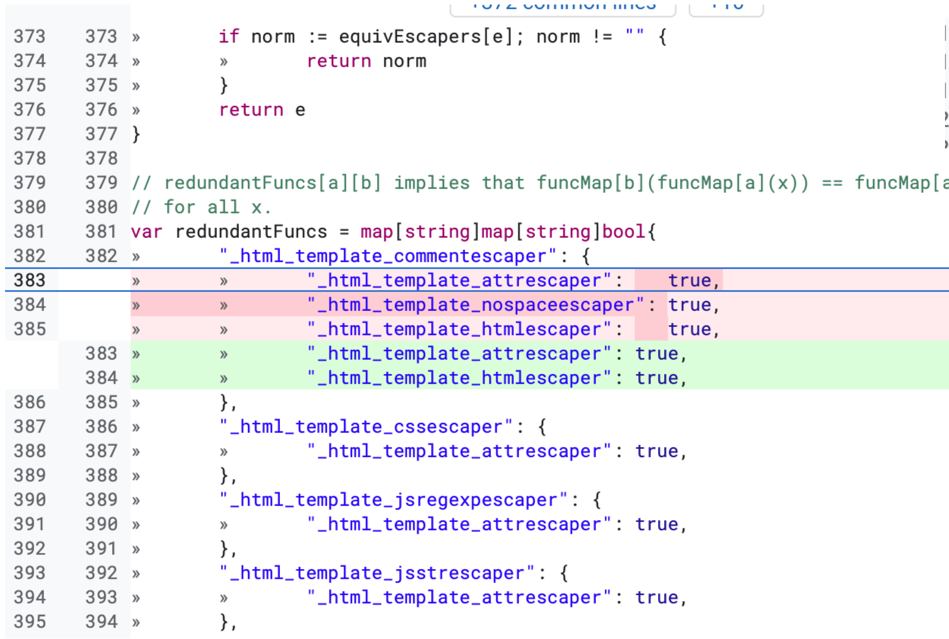
**CVSS 3.x Severity and Metrics:**

**NIST:** NVD **Base Score:** 9.8 CRITICAL

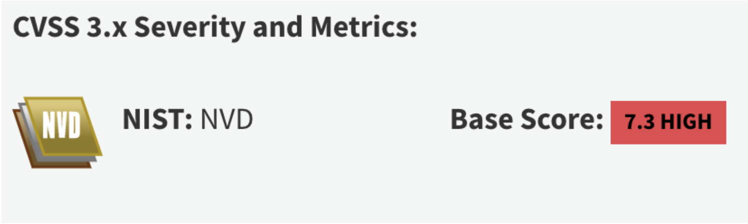
# html/template: improper handling of empty HTML attributes



```
// htmlNospaceEscaper escapes for inclusion in unquoted attribute values.
func htmlNospaceEscaper(args ...any) string {
    s, t := stringify(args...)
    if s == "" {
        return filterFailsafe
    }
    if t == contentTypeHTML {
        return htmlReplacer(stripTags(s), htmlNospaceNormReplacementTable, false)
    }
    return htmlReplacer(s, htmlNospaceReplacementTable, false)
}
```



```
    if norm := equivEscapers[e]; norm != "" {
        return norm
    }
    return e
}

// redundantFuncs[a][b] implies that funcMap[b](funcMap[a](x)) == funcMap[a](x)
// for all x.
var redundantFuncs = map[string]map[string]bool{
    "_html_template_commentescaper": {
        "_html_template_attrescaper":    true,
        "_html_template_nospaceescaper": true,
        "_html_template_htmlescaper":    true,
        "_html_template_attrescaper": true,
        "_html_template_htmlescaper": true,
    },
    "_html_template_cssescaper": {
        "_html_template_attrescaper": true,
    },
    "_html_template_jsregexpescaper": {
        "_html_template_attrescaper": true,
    },
    "_html_template_jsstrescaper": {
        "_html_template_attrescaper": true,
    },
```

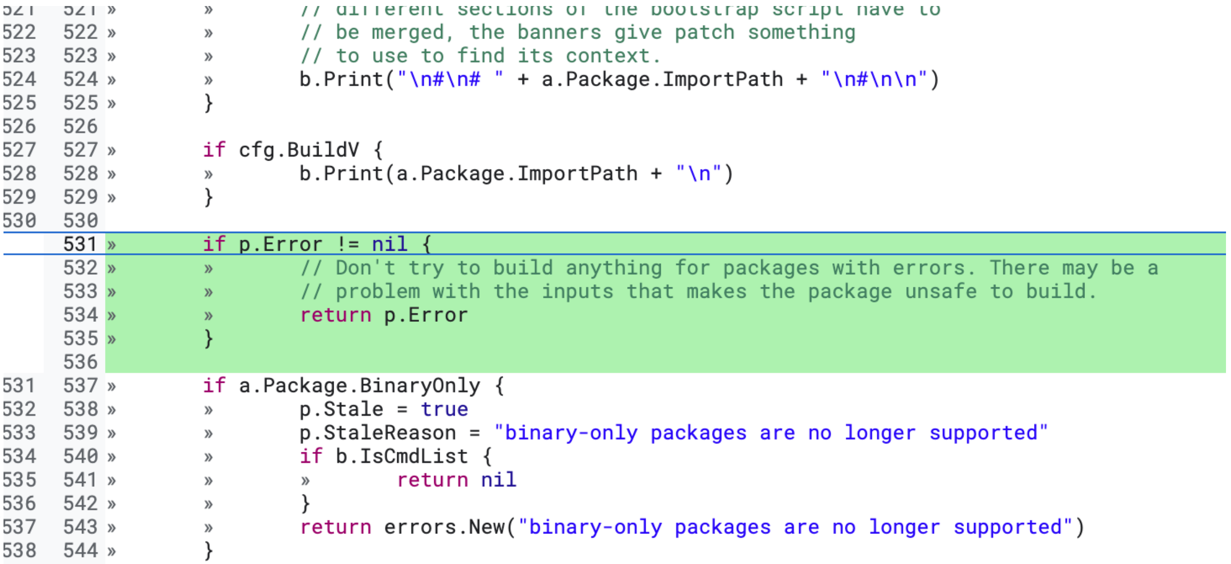


**CVSS 3.x Severity and Metrics:**

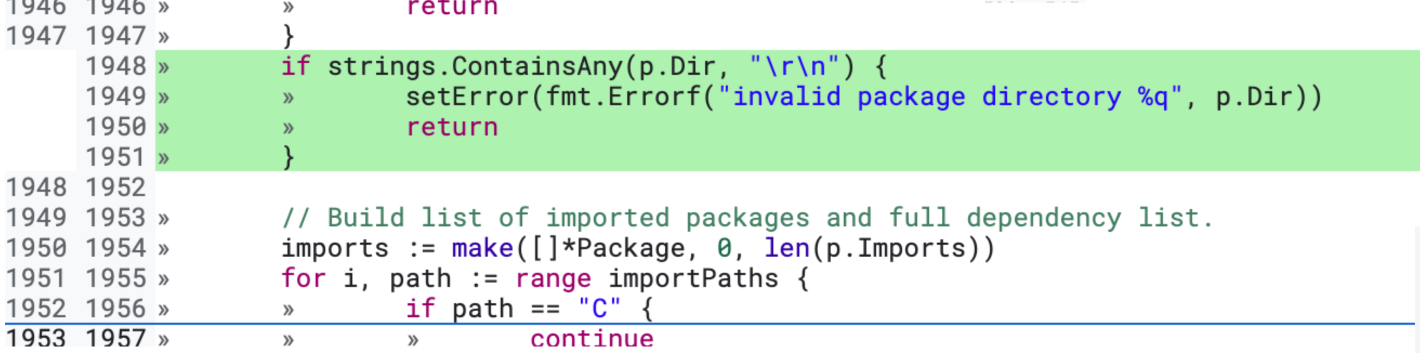
**NIST:** NVD

**Base Score:** 7.3 HIGH

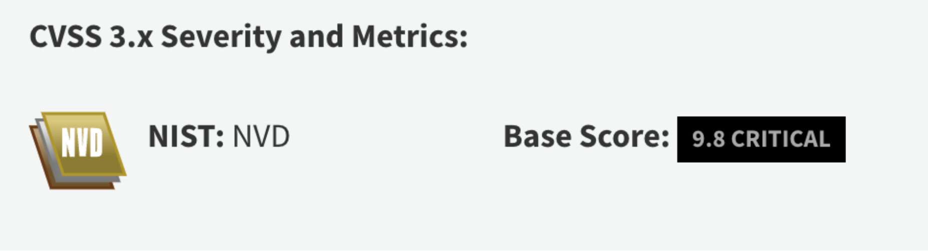
# cmd/go: cgo code injection



```
»       »       // different sections of the bootstrap script have to
»       »       // be merged, the banners give patch something
»       »       // to use to find its context.
»       »       b.Print("\n#\n# " + a.Package.ImportPath + "\n#\n\n")
»       }

»       if cfg.BuildV {
»       »       b.Print(a.Package.ImportPath + "\n")
»       }

»       if p.Error != nil {
»       »       // Don't try to build anything for packages with errors. There may be a
»       »       // problem with the inputs that makes the package unsafe to build.
»       »       return p.Error
»       }

»       if a.Package.BinaryOnly {
»       »       p.Stale = true
»       »       p.StaleReason = "binary-only packages are no longer supported"
»       »       if b.IsCmdList {
»       »       »       return nil
»       »       }
»       »       return errors.New("binary-only packages are no longer supported")
»       }
```



```
»       }
»       if name := pathpkg.Base(p.ImportPath); !SafeA
»       »       setError(fmt.Errorf("invalid input di
»       »       return
»       }
»       if strings.ContainsAny(p.Dir, "\r\n") {
»       »       setError(fmt.Errorf("invalid package directory %q", p.Dir))
»       »       return
»       }

»       // Build list of imported packages and full dependency list.
»       imports := make([]*Package, 0, len(p.Imports))
»       for i, path := range importPaths {
»       »       if path == "C" {
»       »       »       continue
```

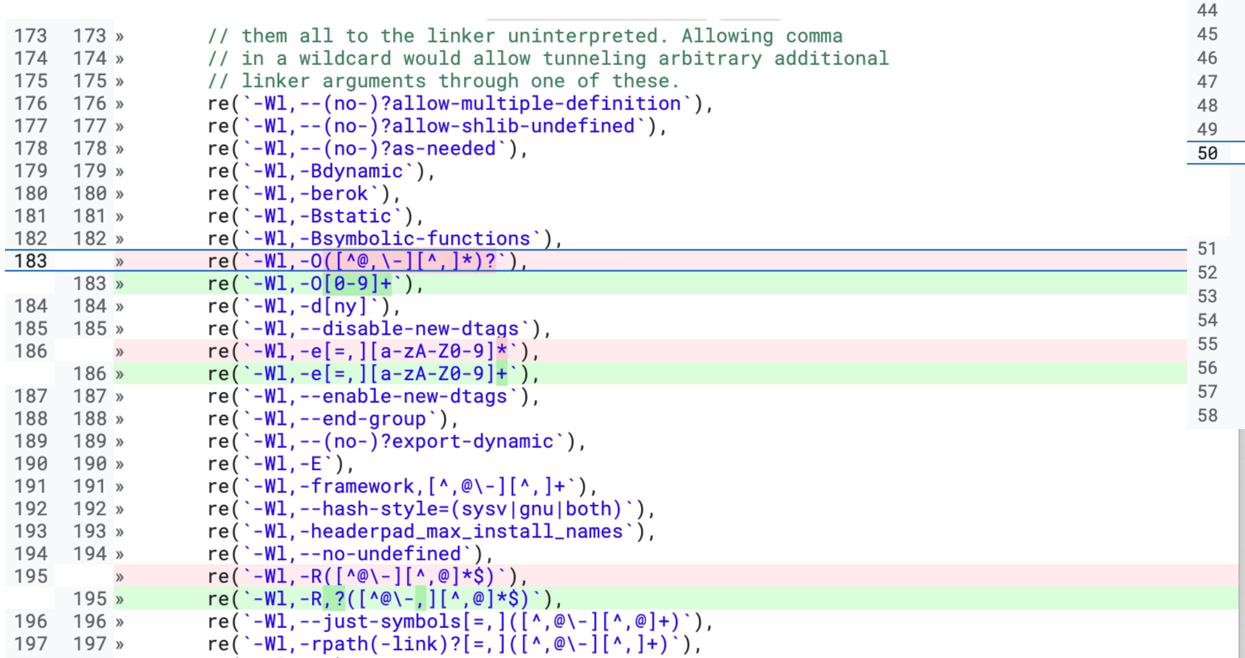


**CVSS 3.x Severity and Metrics:**

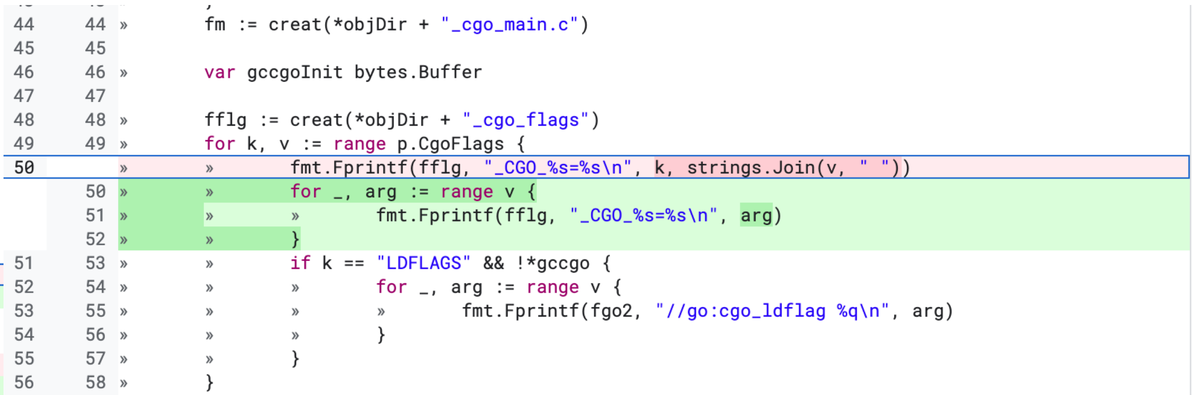
**NIST:** NVD

**Base Score:** 9.8 CRITICAL

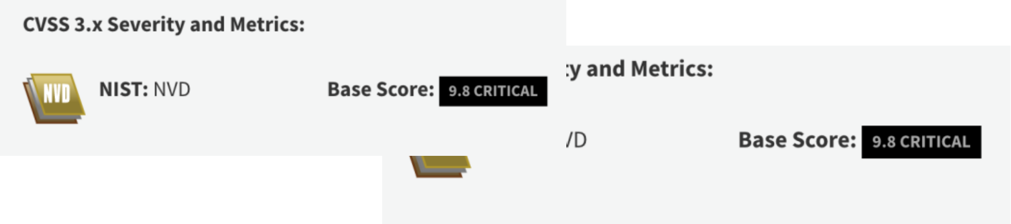
# cmd/go: improper sanitization of LDFLAGS



```
// them all to the linker uninterpreted. Allowing comma
// in a wildcard would allow tunneling arbitrary additional
// linker arguments through one of these.
re(`-Wl,--(no-)?allow-multiple-definition`),
re(`-Wl,--(no-)?allow-shlib-undefined`),
re(`-Wl,--(no-)?as-needed`),
re(`-Wl,-Bdynamic`),
re(`-Wl,-berok`),
re(`-Wl,-Bstatic`),
re(`-Wl,-Bsymbolic-functions`),
re(`-Wl,-O([^@,\-][^,]*)?`),
re(`-Wl,-O[0-9]+`),
re(`-Wl,-d[ny]`),
re(`-Wl,--disable-new-dtags`),
re(`-Wl,-e[=,][a-zA-Z0-9]*`),
re(`-Wl,-e[=,][a-zA-Z0-9]+`),
re(`-Wl,--enable-new-dtags`),
re(`-Wl,--end-group`),
re(`-Wl,--(no-)?export-dynamic`),
re(`-Wl,-E`),
re(`-Wl,-framework,[^,@\-][^,]+`),
re(`-Wl,--hash-style=(sysv|gnu|both)`),
re(`-Wl,-headerpad_max_install_names`),
re(`-Wl,--no-undefined`),
re(`-Wl,-R([^@\-][^,@]*$)`),
re(`-Wl,-R,?([^@\-,][^,@]*$)`),
re(`-Wl,--just-symbols[=,]([^,@\-][^,@]+)`),
re(`-Wl,-rpath(-link)?[=,]([^,@\-][^,]+)`),
re(`-Wl,-s`),
re(`-Wl,-search_
re(`-Wl,-sectcre
re(`-Wl,--start-
re(`-Wl,-?-stati
re(`-Wl,-?-subsy
re(`-Wl,-syslibr
re(`-Wl,-undefin
```



```
fm := creat(*objDir + "_cgo_main.c")

var gccgoInit bytes.Buffer

fflg := creat(*objDir + "_cgo_flags")
for k, v := range p.CgoFlags {
        fmt.Fprintf(fflg, "_CGO_%s=%s\n", k, strings.Join(v, " "))
        for _, arg := range v {
                fmt.Fprintf(fflg, "_CGO_%s=%s\n", arg)
        }
        if k == "LDFLAGS" && !*gccgo {
                for _, arg := range v {
                        fmt.Fprintf(fgo2, "//go:cgo_ldflag %q\n", arg)
                }
        }
}
```

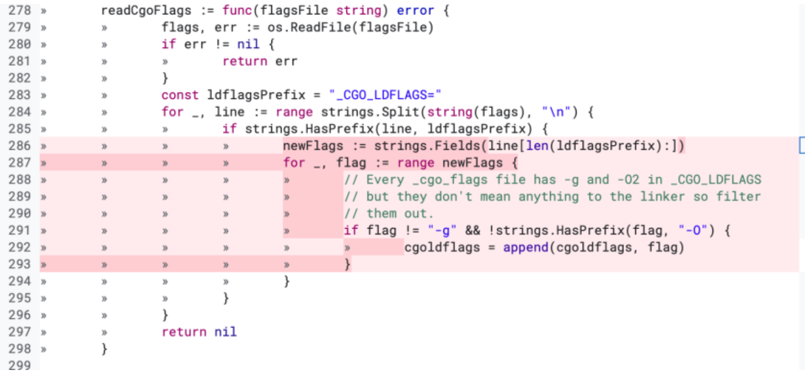


CVSS 3.x Severity and Metrics:

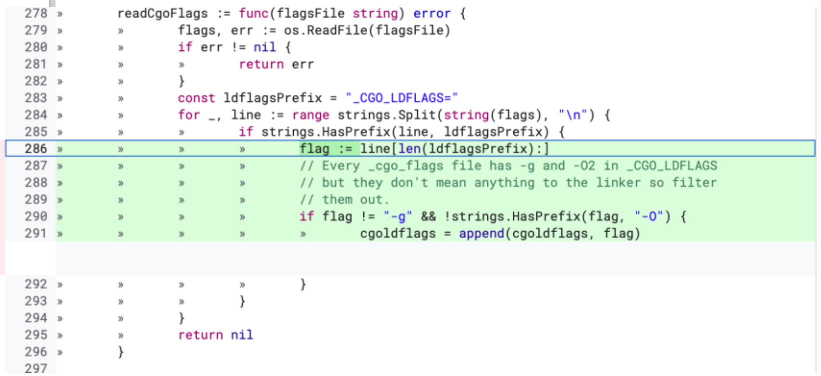
NIST: NVD Base Score: 9.8 CRITICAL

y and Metrics:

/D Base Score: 9.8 CRITICAL

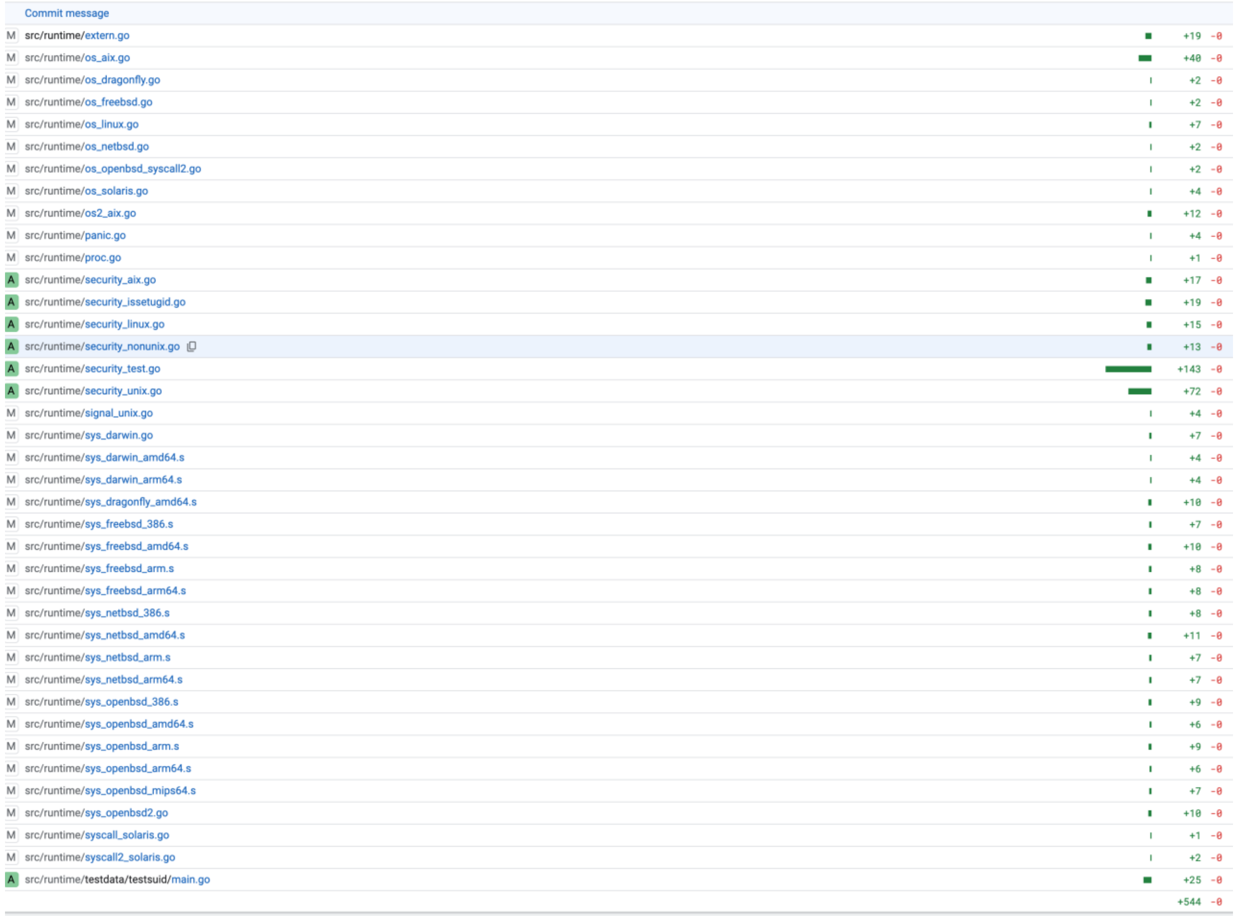


```
readCgoFlags := func(flagsFile string) error {
        flags, err := os.ReadFile(flagsFile)
        if err != nil {
                return err
        }
        const ldflagsPrefix = "_CGO_LDFLAGS="
        for _, line := range strings.Split(string(flags), "\n") {
                if strings.HasPrefix(line, ldflagsPrefix) {
                        newFlags := strings.Fields(line[len(ldflagsPrefix):])
                        for _, flag := range newFlags {
                                // Every _cgo_flags file has -g and -O2 in _CGO_LDFLAGS
                                // but they don't mean anything to the linker so filter
                                // them out.
                                if flag != "-g" && !strings.HasPrefix(flag, "-O") {
                                        cgoldflags = append(cgoldflags, flag)
                                }
                        }
                }
        }
        return nil
}
```

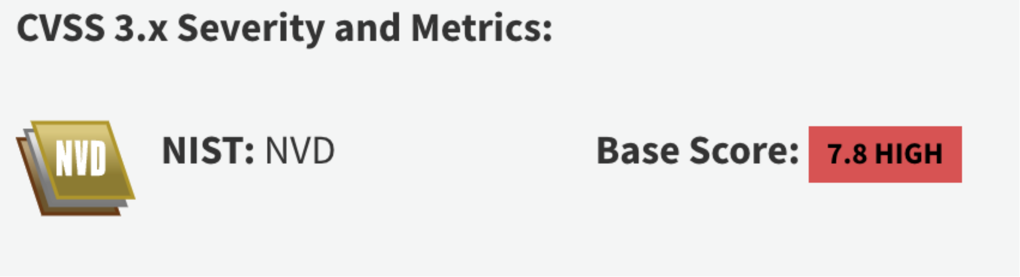


```
readCgoFlags := func(flagsFile string) error {
        flags, err := os.ReadFile(flagsFile)
        if err != nil {
                return err
        }
        const ldflagsPrefix = "_CGO_LDFLAGS="
        for _, line := range strings.Split(string(flags), "\n") {
                if strings.HasPrefix(line, ldflagsPrefix) {
                        flag := line[len(ldflagsPrefix):]
                        // Every _cgo_flags file has -g and -O2 in _CGO_LDFLAGS
                        // but they don't mean anything to the linker so filter
                        // them out.
                        if flag != "-g" && !strings.HasPrefix(flag, "-O") {
                                cgoldflags = append(cgoldflags, flag)
                        }
                }
        }
        return nil
}
```

# runtime: unexpected behavior of setuid/setgid binaries



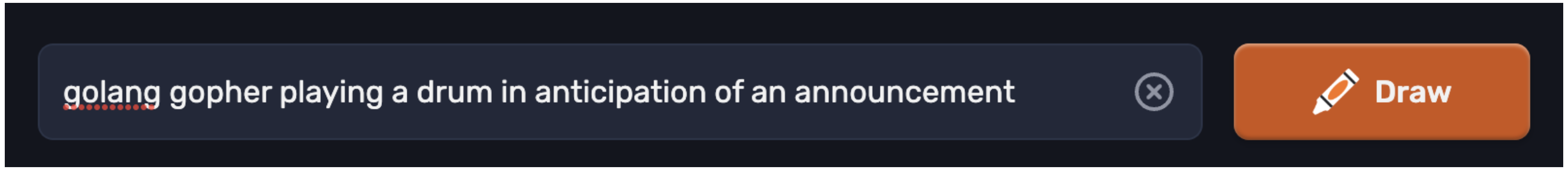
| | | | |
|---|---|---|---|
| | Commit message | | |
| M | src/runtime/extern.go | +19 | -0 |
| M | src/runtime/os_aix.go | +40 | -0 |
| M | src/runtime/os_dragonfly.go | +2 | -0 |
| M | src/runtime/os_freebsd.go | +2 | -0 |
| M | src/runtime/os_linux.go | +7 | -0 |
| M | src/runtime/os_netbsd.go | +2 | -0 |
| M | src/runtime/os_openbsd_syscall2.go | +2 | -0 |
| M | src/runtime/os_solaris.go | +4 | -0 |
| M | src/runtime/os2_aix.go | +12 | -0 |
| M | src/runtime/panic.go | +4 | -0 |
| M | src/runtime/proc.go | +1 | -0 |
| A | src/runtime/security_aix.go | +17 | -0 |
| A | src/runtime/security_issetugid.go | +19 | -0 |
| A | src/runtime/security_linux.go | +15 | -0 |
| A | src/runtime/security_nonunix.go | +13 | -0 |
| A | src/runtime/security_test.go | +143 | -0 |
| A | src/runtime/security_unix.go | +72 | -0 |
| M | src/runtime/signal_unix.go | +4 | -0 |
| M | src/runtime/sys_darwin.go | +7 | -0 |
| M | src/runtime/sys_darwin_amd64.s | +4 | -0 |
| M | src/runtime/sys_darwin_arm64.s | +4 | -0 |
| M | src/runtime/sys_dragonfly_amd64.s | +10 | -0 |
| M | src/runtime/sys_freebsd_386.s | +7 | -0 |
| M | src/runtime/sys_freebsd_amd64.s | +10 | -0 |
| M | src/runtime/sys_freebsd_arm.s | +8 | -0 |
| M | src/runtime/sys_freebsd_arm64.s | +8 | -0 |
| M | src/runtime/sys_netbsd_386.s | +8 | -0 |
| M | src/runtime/sys_netbsd_amd64.s | +11 | -0 |
| M | src/runtime/sys_netbsd_arm.s | +7 | -0 |
| M | src/runtime/sys_netbsd_arm64.s | +7 | -0 |
| M | src/runtime/sys_openbsd_386.s | +9 | -0 |
| M | src/runtime/sys_openbsd_amd64.s | +6 | -0 |
| M | src/runtime/sys_openbsd_arm.s | +9 | -0 |
| M | src/runtime/sys_openbsd_arm64.s | +6 | -0 |
| M | src/runtime/sys_openbsd_mips64.s | +7 | -0 |
| M | src/runtime/sys_openbsd2.go | +10 | -0 |
| M | src/runtime/syscall_solaris.go | +1 | -0 |
| M | src/runtime/syscall2_solaris.go | +2 | -0 |
| A | src/runtime/testdata/testsuid/main.go | +25 | -0 |
| | | +544 | -0 |



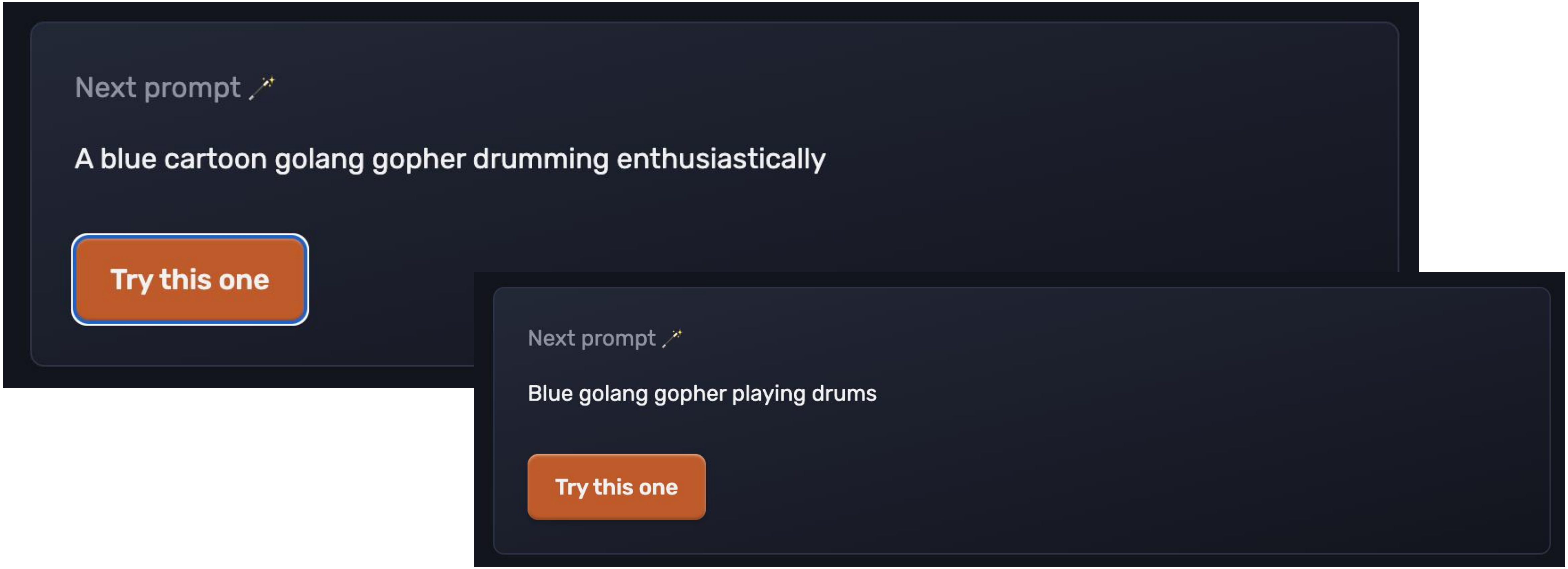
**CVSS 3.x Severity and Metrics:**

**NIST:** NVD

**Base Score:** 7.8 HIGH



golang gopher playing a drum in anticipation of an announcement



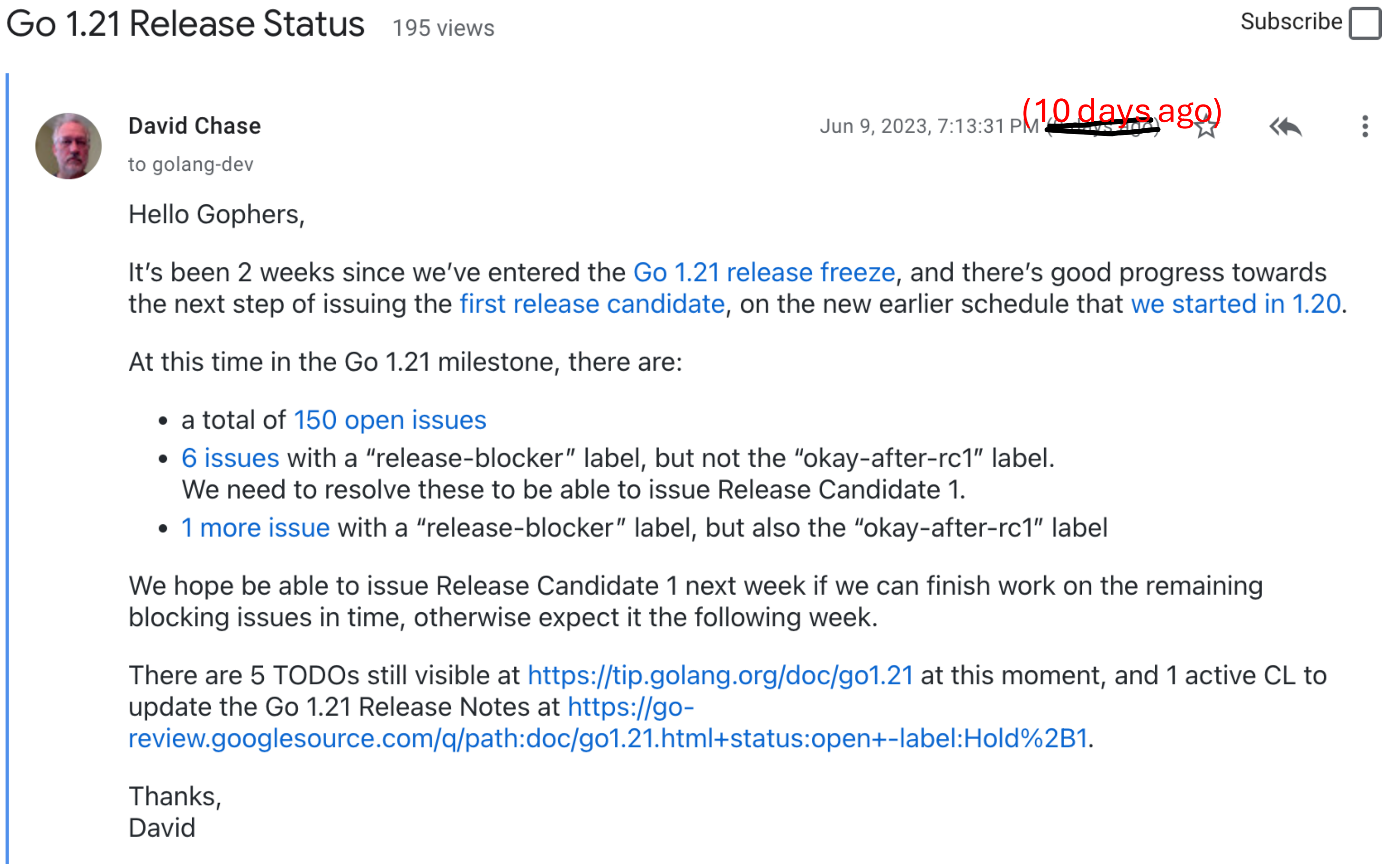
Next prompt 🪄

A blue cartoon golang gopher drumming enthusiastically

Next prompt 🪄

Blue golang gopher playing drums

# go1.21rc1



Go 1.21 Release Status 195 views

**David Chase**

to golang-dev

Jun 9, 2023, 7:13:31 PM [redacted] (10 days ago)

Hello Gophers,

It’s been 2 weeks since we’ve entered the Go 1.21 release freeze, and there’s good progress towards the next step of issuing the first release candidate, on the new earlier schedule that we started in 1.20.

At this time in the Go 1.21 milestone, there are:

- a total of 150 open issues
- 6 issues with a “release-blocker” label, but not the “okay-after-rc1” label.
  We need to resolve these to be able to issue Release Candidate 1.
- 1 more issue with a “release-blocker” label, but also the “okay-after-rc1” label

We hope be able to issue Release Candidate 1 next week if we can finish work on the remaining blocking issues in time, otherwise expect it the following week.

There are 5 TODOs still visible at https://tip.golang.org/doc/go1.21 at this moment, and 1 active CL to update the Go 1.21 Release Notes at https://go-review.googlesource.com/q/path:doc/go1.21.html+status:open+-label:Hold%2B1.

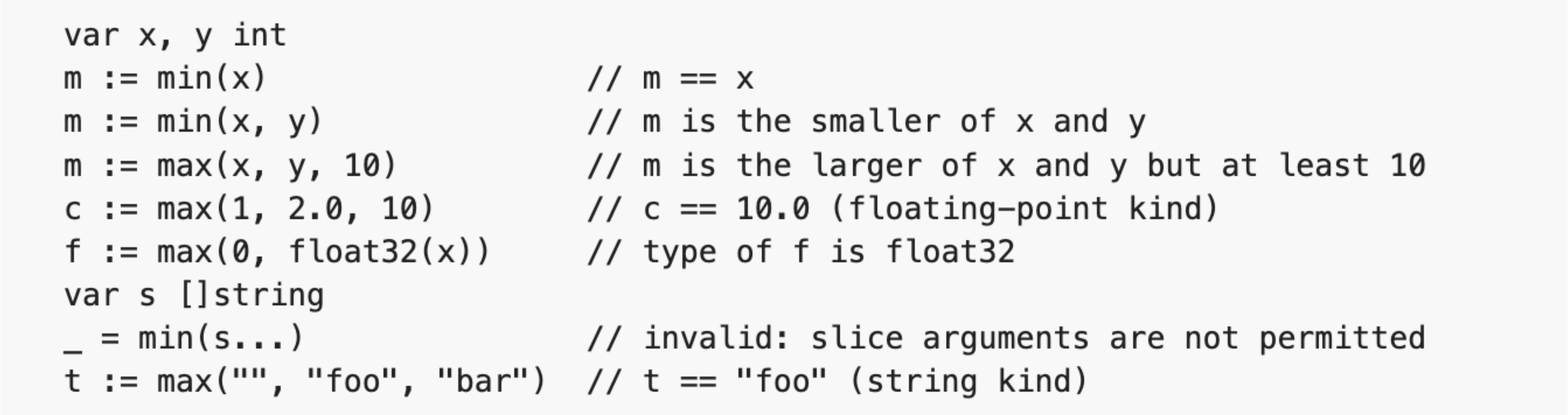
Thanks,
David

# New builtins

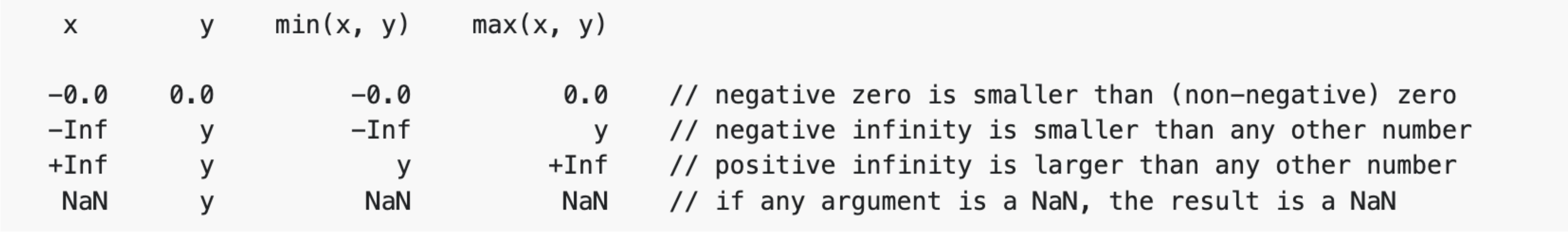
Go 1.21 adds three new built-ins to the language.

- `min`
- `max`
- `clear`

# New builtins – min/max

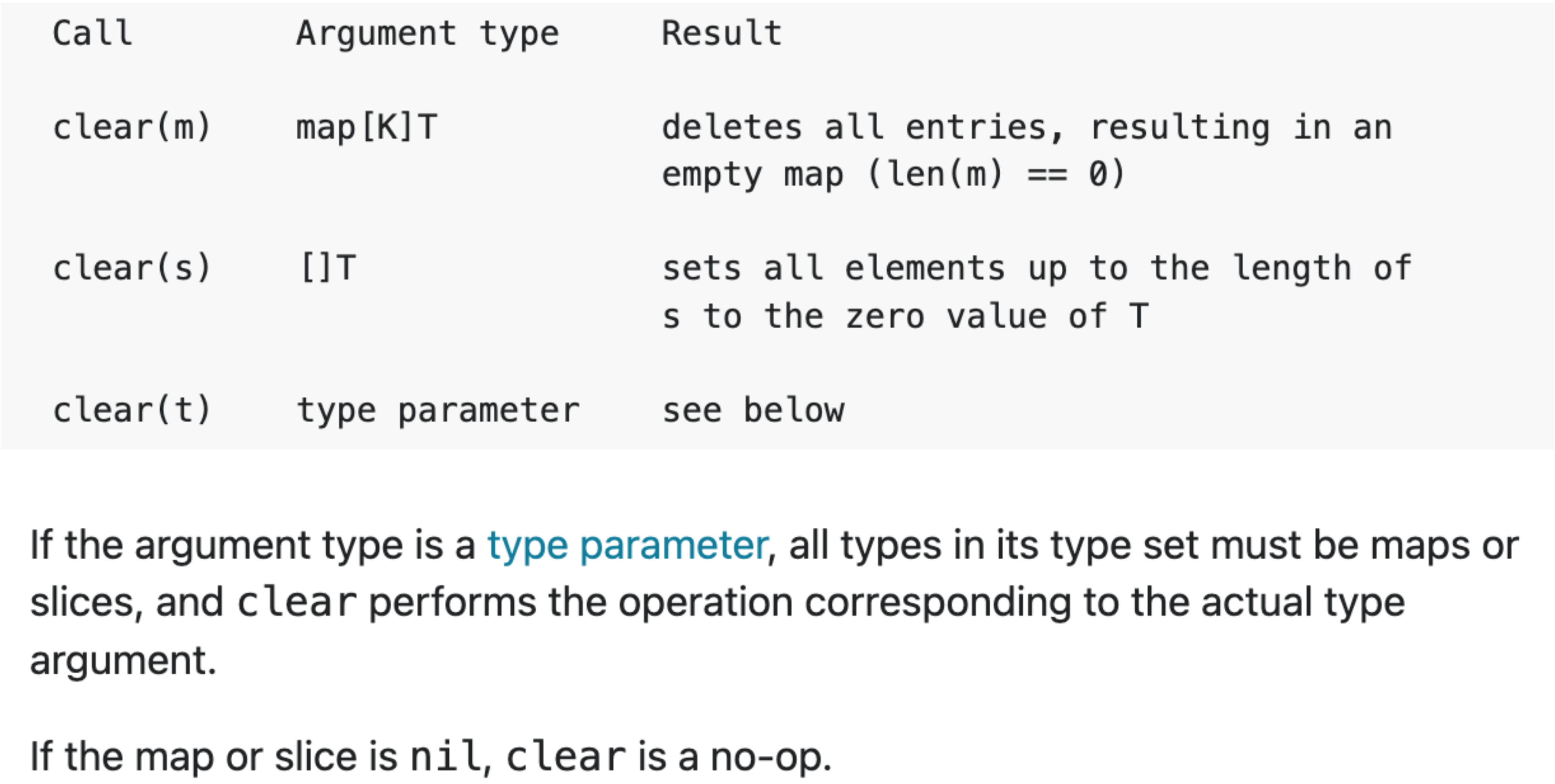


```
var x, y int
m := min(x)                 // m == x
m := min(x, y)              // m is the smaller of x and y
m := max(x, y, 10)          // m is the larger of x and y but at least 10
c := max(1, 2.0, 10)        // c == 10.0 (floating-point kind)
f := max(0, float32(x))     // type of f is float32
var s []string
_ = min(s...)               // invalid: slice arguments are not permitted
t := max("", "foo", "bar")  // t == "foo" (string kind)
```



```
 x        y    min(x, y)    max(x, y)

-0.0    0.0         -0.0          0.0    // negative zero is smaller than (non-negative) zero
-Inf      y         -Inf            y    // negative infinity is smaller than any other number
+Inf      y            y         +Inf    // positive infinity is larger than any other number
 NaN      y          NaN          NaN    // if any argument is a NaN, the result is a NaN
```

# New builtins – clear



| Call | Argument type | Result |
|---|---|---|
| clear(m) | map[K]T | deletes all entries, resulting in an empty map (len(m) == 0) |
| clear(s) | []T | sets all elements up to the length of s to the zero value of T |
| clear(t) | type parameter | see below |

If the argument type is a type parameter, all types in its type set must be maps or slices, and `clear` performs the operation corresponding to the actual type argument.

If the map or slice is `nil`, `clear` is a no-op.

# Clearly defined import order

Package initialization order is now specified more precisely. The new algorithm is:

- Sort all packages by import path.
- Repeat until the list of packages is empty:
  - Find the first package in the list for which all imports are already initialized.
  - Initialize that package and remove it from the list.

# Better Windows debugging

On Windows AMD64, the linker (with help from the compiler) now emits SEH unwinding data by default, which improves the integration of Go applications with Windows debuggers and other tools.

# Go Runtime

- Deep stacktraces show 50 first + 50 last
- Better huge page support on Linux
- Better GC (or not)
- Improved C to  Go calls on thread created in C ¯_(ツ)_/¯

# Better compatibility – Backward & Forward

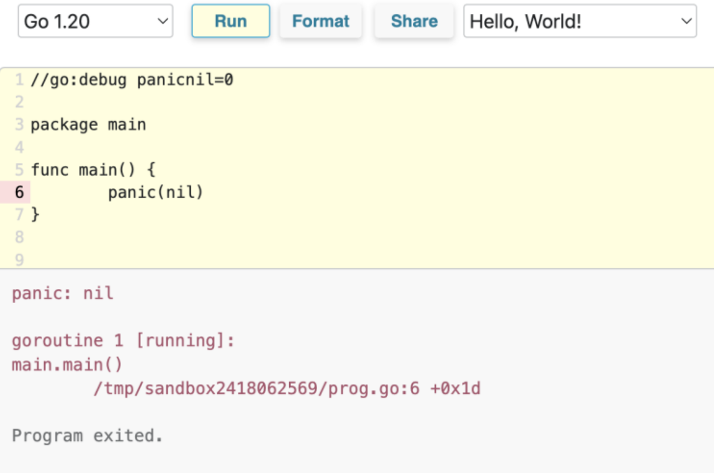
- Programs relying on a bug in the toolchain

- GODEBUG=…
  - key=value,…
  - Kind of a featureflag for breaking changes in the compiler toolchain

- Go1.21 introduces the following default resolution
  - Default per toolchain
  - Default for version in go.mod / go.work
  - `//go:debug lines`

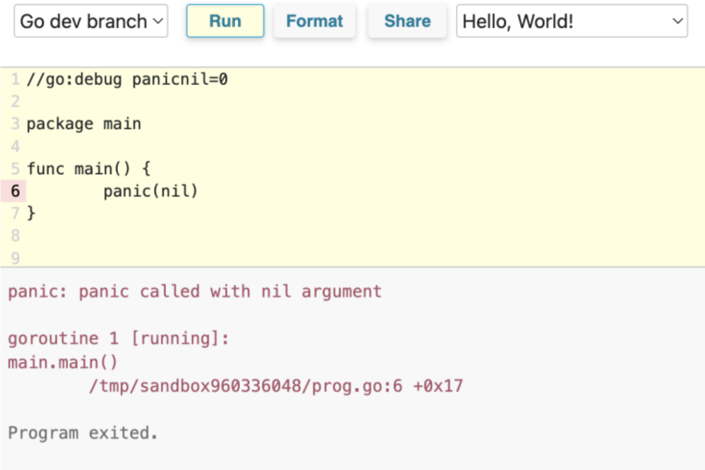
# Better compatibility – Backward & Forward

- Go1.21 introduces `panicnil=0` for `GODEBUG`



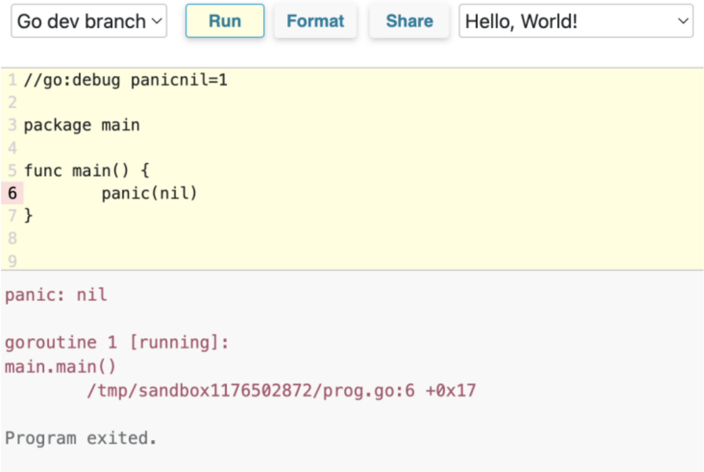
```
//go:debug panicnil=0

package main

func main() {
	panic(nil)
}
```

```
panic: nil

goroutine 1 [running]:
main.main()
	/tmp/sandbox2418062569/prog.go:6 +0x1d

Program exited.
```



```
//go:debug panicnil=0

package main

func main() {
	panic(nil)
}
```

```
panic: panic called with nil argument

goroutine 1 [running]:
main.main()
	/tmp/sandbox960336048/prog.go:6 +0x17

Program exited.
```



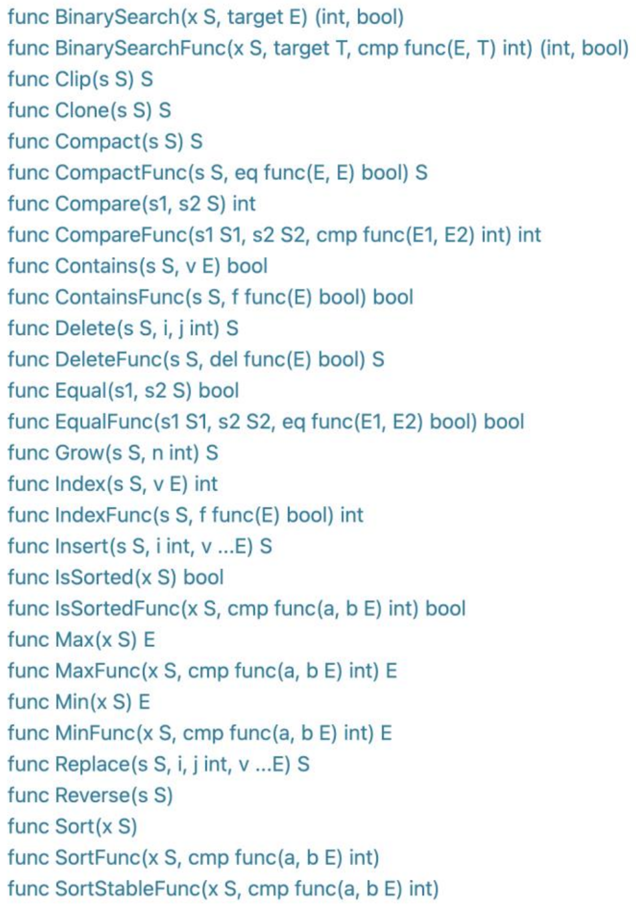
```
//go:debug panicnil=1

package main

func main() {
	panic(nil)
}
```

```
panic: nil

goroutine 1 [running]:
main.main()
	/tmp/sandbox1176502872/prog.go:6 +0x17

Program exited.
```

# New packages

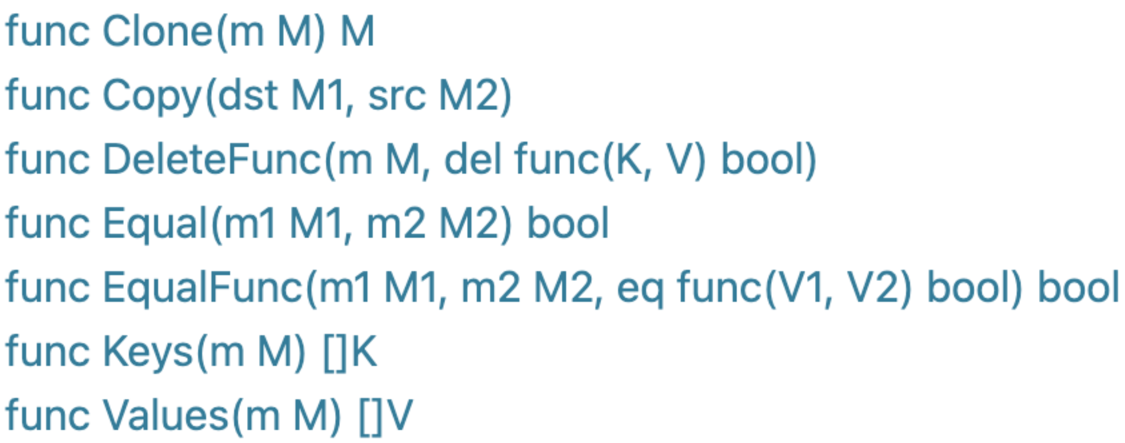
- log/slog
- slices
- maps
- cmp

# New packages - slices



```
func BinarySearch(x S, target E) (int, bool)
func BinarySearchFunc(x S, target T, cmp func(E, T) int) (int, bool)
func Clip(s S) S
func Clone(s S) S
func Compact(s S) S
func CompactFunc(s S, eq func(E, E) bool) S
func Compare(s1, s2 S) int
func CompareFunc(s1 S1, s2 S2, cmp func(E1, E2) int) int
func Contains(s S, v E) bool
func ContainsFunc(s S, f func(E) bool) bool
func Delete(s S, i, j int) S
func DeleteFunc(s S, del func(E) bool) S
func Equal(s1, s2 S) bool
func EqualFunc(s1 S1, s2 S2, eq func(E1, E2) bool) bool
func Grow(s S, n int) S
func Index(s S, v E) int
func IndexFunc(s S, f func(E) bool) int
func Insert(s S, i int, v ...E) S
func IsSorted(x S) bool
func IsSortedFunc(x S, cmp func(a, b E) int) bool
func Max(x S) E
func MaxFunc(x S, cmp func(a, b E) int) E
func Min(x S) E
func MinFunc(x S, cmp func(a, b E) int) E
func Replace(s S, i, j int, v ...E) S
func Reverse(s S)
func Sort(x S)
func SortFunc(x S, cmp func(a, b E) int)
func SortStableFunc(x S, cmp func(a, b E) int)
```

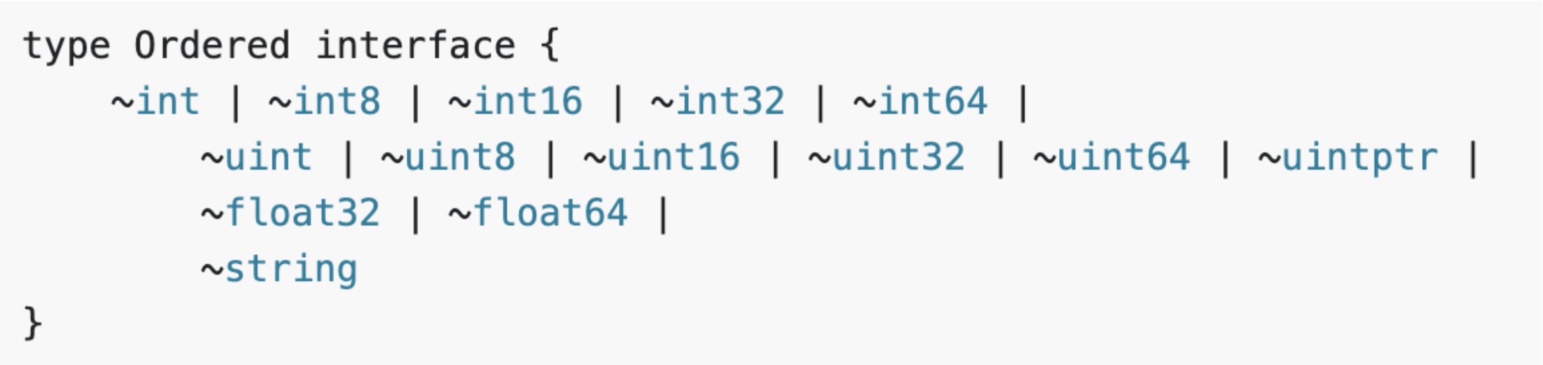
# New packages - maps



```
func Clone(m M) M
func Copy(dst M1, src M2)
func DeleteFunc(m M, del func(K, V) bool)
func Equal(m1 M1, m2 M2) bool
func EqualFunc(m1 M1, m2 M2, eq func(V1, V2) bool) bool
func Keys(m M) []K
func Values(m M) []V
```

# New packages - cmp

func Compare(x, y T) int
func Less(x, y T) bool
type Ordered



```
type Ordered interface {
    ~int | ~int8 | ~int16 | ~int32 | ~int64 |
        ~uint | ~uint8 | ~uint16 | ~uint32 | ~uint64 | ~uintptr |
        ~float32 | ~float64 |
        ~string
}
```

any type that supports the operators < <= >= >. If future releases of Go add new ordered types, this constraint will be modified to include them.

# Changes to packages

- context
  - WithoutCancel
  - WithDeadlineCause
  - WithTimeoutCause
  - AfterFunc

- crypto/rsa
  - Fixed a performance regression from go1.20 (both `amd64` and `arm64`)

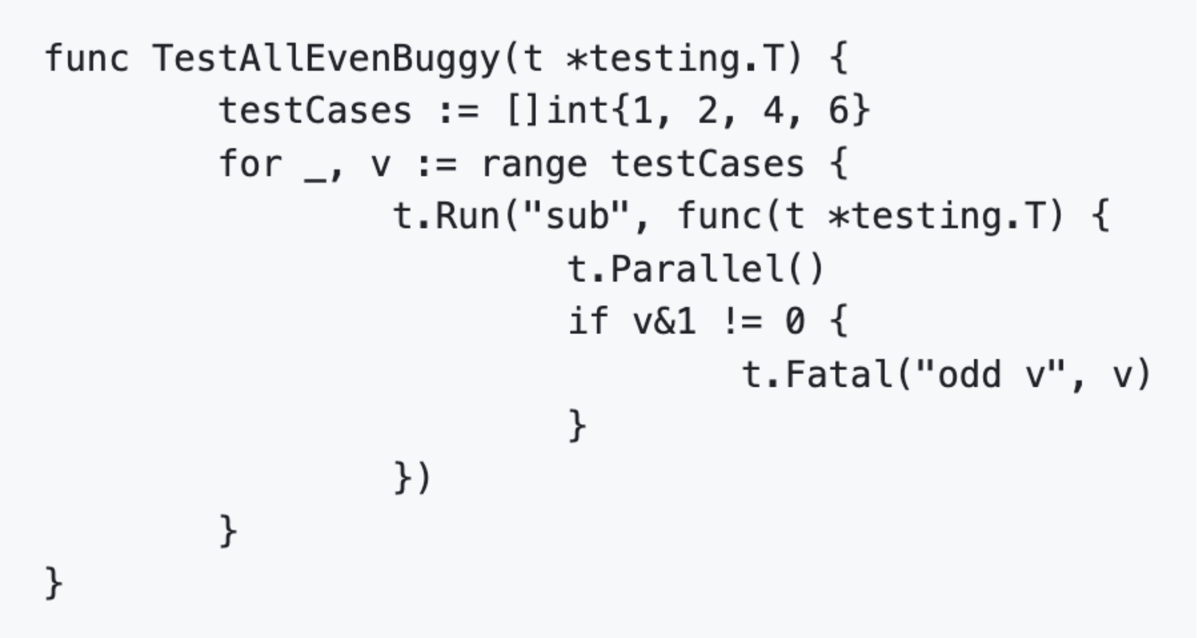
- crypto/sha256
  - Uses native CPU instructions for amd64 (3-4x speed improvement)

# Changes to packages

- unicode
  - upgraded to unicode 15.0.0

© 2022 Emojipedia Sample Image Collection for draft Emoji 15.0. Final designs will vary by vendor.

# Loopvar Experiment



```
func TestAllEvenBuggy(t *testing.T) {
        testCases := []int{1, 2, 4, 6}
        for _, v := range testCases {
                t.Run("sub", func(t *testing.T) {
                        t.Parallel()
                        if v&1 != 0 {
                                t.Fatal("odd v", v)
                        }
                })
        }
}
```

```
GOEXPERIMENT=loopvar go build my/program
```