

Go at Tailscale

Brad Fitzpatrick
FOSDEM, Feb 2021

Hello again!

Long time, no see!



FOSDEM¹⁵



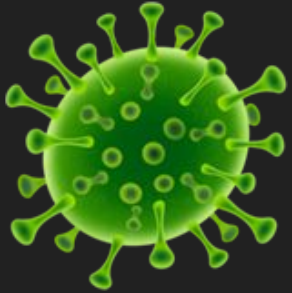
Go

K.3.401



Conference? Nope!

Ball!



Hello for the
first time!!



LiveJournal, memcached, OpenID, ...

a wise friend, Evan



You should try Debian.

- 1999



I reimplemented git in
Ocaml and Haskell and
have opinions.



Rust looks like it will be interesting.

- 2006/2007-ish



No more Linux.
I only use Windows
now.
Won't tell you why.





LiveJournal, memcached, OpenID, ...



Google

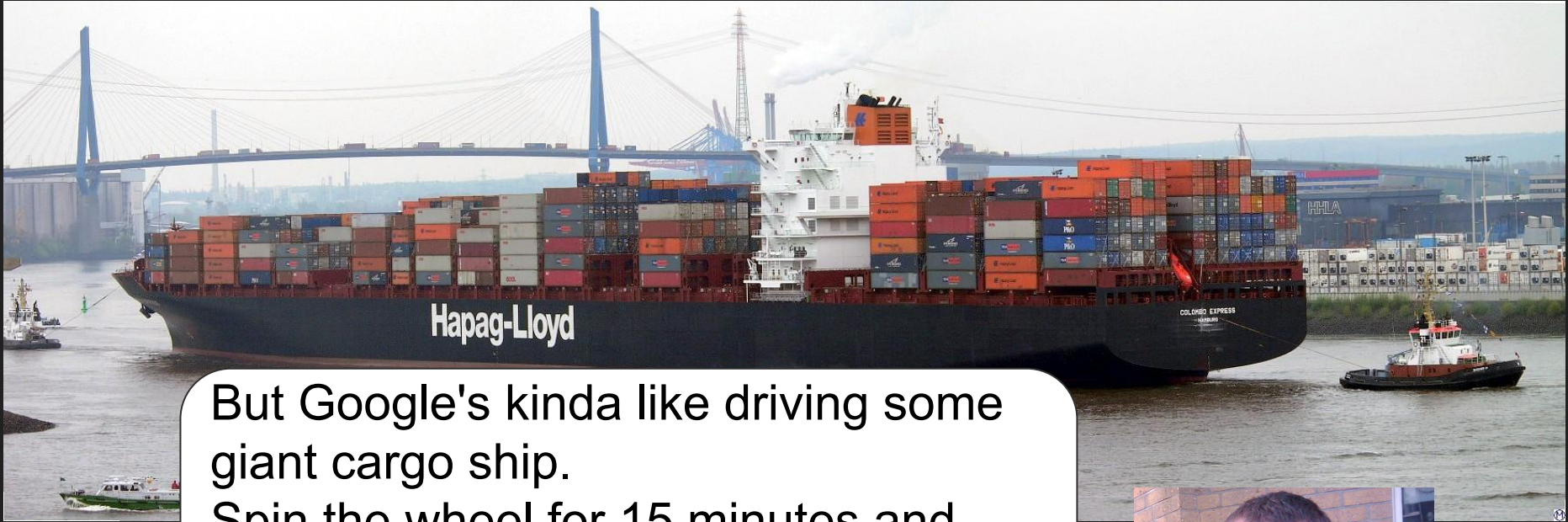


A man with a beard and sunglasses, wearing a blue cap, is jet skiing on a body of water. He is smiling and holding the handlebars. A speech bubble is positioned to his right, containing text. In the bottom right corner, there is a small inset image of a man with a beard and a grey scarf.

Welcome to Google!

... a start-up is kinda
like driving a jet-ski.





But Google's kinda like driving some giant cargo ship.
Spin the wheel for 15 minutes and nothing happens. But that momentum when it does! And those resources!



LiveJournal, memcached, OpenID, ...



Google (social, gmail BE, Android)



Go for 10 yrs (std, http, CI, ...)



....

But time for a change!





What is Tailscale?

Tailscale

- "new style" VPN
 - we don't run any "exit nodes" (you can)
 - end-to-end encryption, SSO identity
- you can talk with peers in your network (your or your organization's devices)
 - or share devices with friends/coworkers
- WireGuard + SSO + key management + ACLs + NAT/firewall traversal
 - "mesh" VPN: peers communicate directly without HW or SW "concentrator"
- gives each of your devices a **magic IP addresses that Just Works** between any OS, any device, any network, without network config
- open source: non-GUI client, Android app, core
 - others beat us to releasing open source server component
- free for individuals
 - we make money from companies

Any OS?

Linux

Windows

macOS

iOS

Android

FreeBSD, pfSense, OPNsense

Synology

Ubiquiti EdgeRouter

....

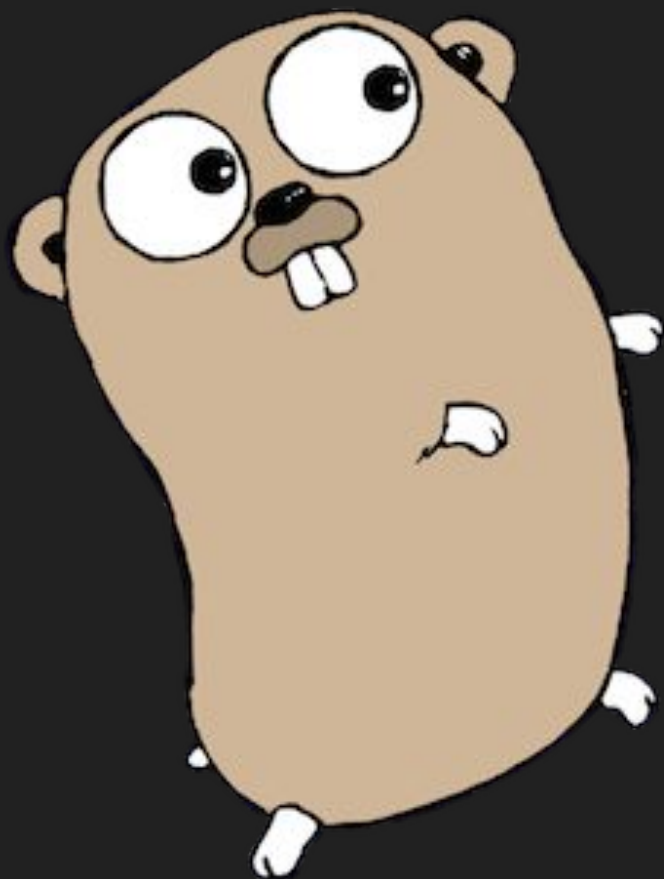


If only there was a way to write
cross-platform native applications!

Electron!



Naah





Linux



Windows



macOS



iOS



Android



FreeBSD, pfSense, OPNsense



Synology



Ubiquiti EdgeRouter

....

Linux

Linux, *BSD, etc

CLI & daemon

go install tailscale.com/cmd/tailscale{,d}
/dev/net/tun

netlink, devd for changes

TUN

inject IP packets in/out of kernel
(Tailscale is all userspace)

plan to use faster userspace interfaces
later

Windows

Auth expires in: 3240 hours

My address: 100.85.149.128

7607604 received | 4998240 sent | 2 links

Connect

Disconnect

☒ Allow Incoming Connections

Run unattended

Admin console...

Network devices >

Log in...

Log out (bradfitz@tailscale.com)

About...

Exit



9:11 PM
1/23/2021



Windows

use wireguard-go's Wintun
tailscaled (OSS)

tailscale.exe CLI (OSS)

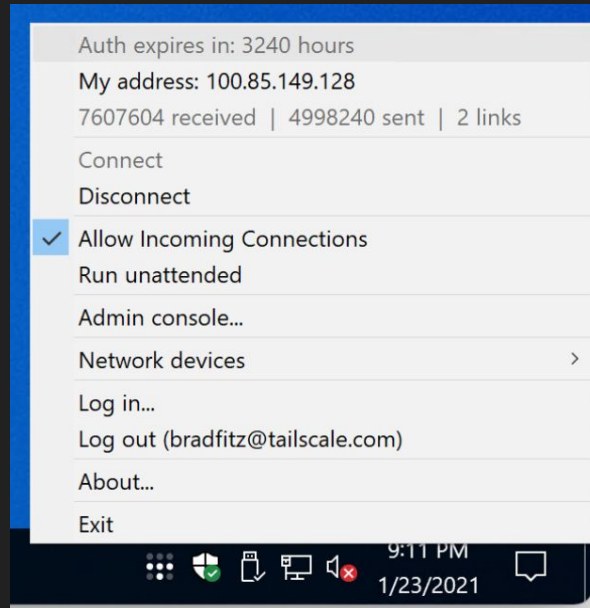
golang.org/x/sys/windows{,svc}
system tray GUI (uses lxn/walk)

```
import ("github.com/lxn/walk" ..)
```

```
...  
    if a.ni, err = walk.NewNotifyIcon(a.mw); err != nil {  
        return err  
    }  
...
```

```
    a.Start = a.addActionLocked("&Connect", false, func() {  
        a.b.UpdatePrefs(func(p *ipn.Prefs) { p.WantRunning = true })  
    })  
    a.Stop = a.addActionLocked("&Disconnect", false, func() {  
        a.b.UpdatePrefs(func(p *ipn.Prefs) { p.WantRunning = false })  
    })  
    a.addSeparatorLocked()
```

```
    a.AllowIncoming = a.addCheckedActionLocked("Allow &Incoming Connections",  
        "ShieldsDown", true,  
        func(p *ipn.Prefs) bool { return !p.ShieldsUp },  
        func(p *ipn.Prefs, v bool) { p.ShieldsUp = !v })
```



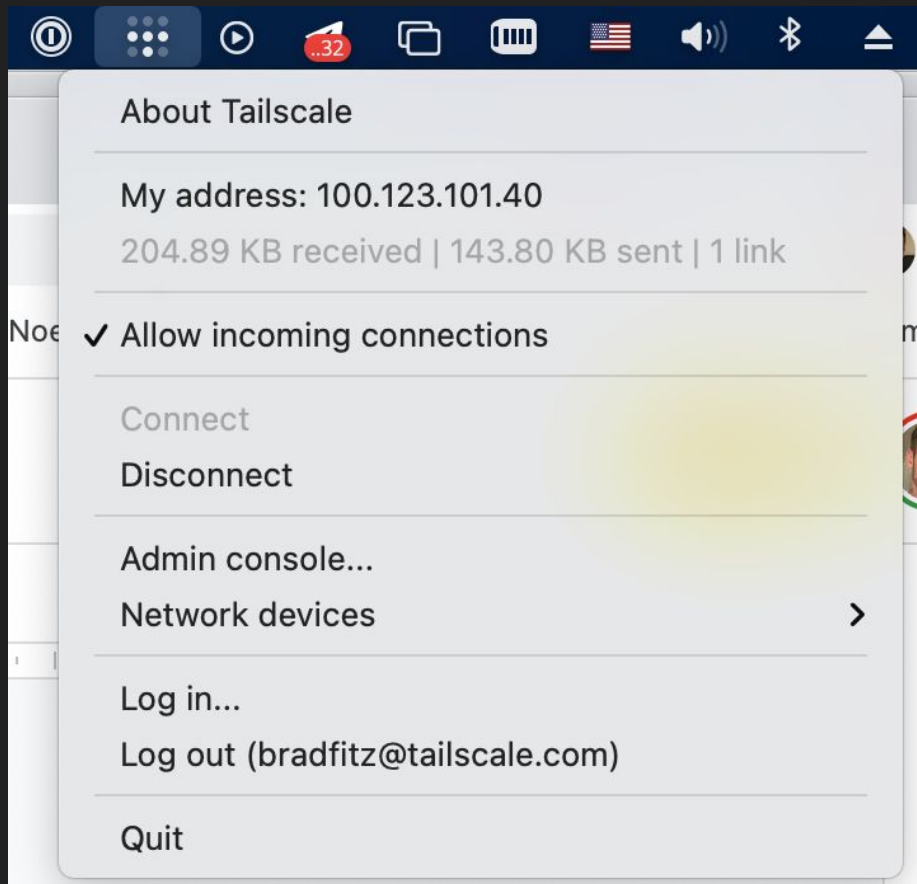
macOS

macOS

go install tailscale.com/cmd/tailscale{,d}

-or-

GUI in App Store



About Tailscale

My address: 100.123.101.40

204.89 KB received | 143.80 KB sent | 1 link

✓ Allow incoming connections

Connect

Disconnect

Admin console...

Network devices >

Log in...

Log out (bradfitz@tailscale.com)

Quit

Go from Swift

```
go build -o $X -buildmode c-archive
```

foo.h

```
extern void goBeCLI(const char * sharedDir);
```

foo.swift

```
func applicationDidFinishLaunching(_ aNotification: Notification) {  
    if let dir = FileManager.default.containerURL(forSecurityApplicationGroupIdentifier:  
FileHelper.appGroupId ?? "") {  
        dir.path.withCString {  
            goBeCLI($0)  
        }  
    }  
}
```

foo.go (the c-archive part)

```
import "C"
```

```
//export goBeCLI
```

```
func goBeCLI(sharedDirStr unsafe.Pointer) {  
    ...  
    if err := cli.Run(os.Args[1:]); err != nil {  
        log.Fatal(err)  
    }  
    os.Exit(0)  
}
```


github.com/progrum/macdriver ??

MacDriver



Native Mac APIs for Golang!

[godoc](#)[reference](#)[test](#)[passing](#)[go report](#)[A](#)[twitter](#)[@progrumHQ](#)[community](#)[forum](#)[sponsor](#)

MacDriver is a toolkit for working with Apple/Mac APIs and frameworks in Go. It currently has 3 "layers":

1. Bindings for Objective-C

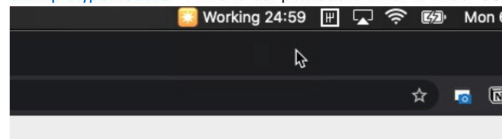
github.com/progrium/macdriver !!

Examples

[examples/largetype](#) - A Contacts/Quicksilver-style Large Type utility in under 80 lines:

[illegible]

[examples/pomodoro](#) - A menu bar pomodoro timer in under 80 lines:



iOS

iOS

Almost identical to macOS codebase
Not open source

Android

Android

github.com/tailscale/tailscale-android

Gio (gioui.org) + JNI

Limited Java

```

__attribute__((visibility ("hidden"))) void _jni_CallVoidMethodA(JNIEnv *env, jobject obj, j
__attribute__((visibility ("hidden"))) jbyteArray _jni_NewByteArray(JNIEnv *env, jsize lengt
__attribute__((visibility ("hidden"))) jbyte *_jni_GetByteArrayElements(JNIEnv *env, jbyteAr
__attribute__((visibility ("hidden"))) void _jni_ReleaseByteArrayElements(JNIEnv *env, jbyte
__attribute__((visibility ("hidden"))) jsize _jni_GetArrayLength(JNIEnv *env, jarray arr);
*/
import "C"

type JVM struct {
    jvm *C.JavaVM
}

type Env struct {
    env *C.JNIEnv
}

type (
    Class      C.jclass
    Object      C.jobject
    MethodID    C.jmethodID
    String      C.jstring
    ByteArray   C.jbyteArray
    Boolean     C.jboolean
    Value       uint64 // All JNI types fit into 64-bits.
)

const (
    True  Boolean = C.JNI_TRUE
    False Boolean = C.JNI_FALSE
)

func JVMFor(jvmPtr uintptr) JVM {
    return JVM{
        jvm: (*C.JavaVM)(unsafe.Pointer(jvmPtr)),
    }
}

func EnvFor(envPtr uintptr) Env {
    return Env{
        env: (*C.JNIEnv)(unsafe.Pointer(envPtr)),
    }
}

```

```

func IsSameObject(e Env, ref1, ref2 Object) bool {
    same := C._jni_IsSameObject(e.env, C.jobject(ref1), C.jobject(ref2))
    return same == C.JNI_TRUE
}

func CallStaticIntMethod(e Env, cls Class, method MethodID, args ...Value) (int, error) {
    res := C._jni_CallStaticIntMethodA(e.env, C.jclass(cls), C.jmethodID(method), varArgs(args))
    return int(res), exception(e)
}

func CallStaticVoidMethod(e Env, cls Class, method MethodID, args ...Value) error {
    C._jni_CallStaticVoidMethodA(e.env, C.jclass(cls), C.jmethodID(method), varArgs(args))
    return exception(e)
}

func CallVoidMethod(e Env, obj Object, method MethodID, args ...Value) error {
    C._jni_CallVoidMethodA(e.env, C.jobject(obj), C.jmethodID(method), varArgs(args))
    return exception(e)
}

func CallStaticObjectMethod(e Env, cls Class, method MethodID, args ...Value) (Object, error) {
    res := C._jni_CallStaticObjectMethodA(e.env, C.jclass(cls), C.jmethodID(method), varArgs(args))
    return Object(res), exception(e)
}

func CallObjectMethod(e Env, obj Object, method MethodID, args ...Value) (Object, error) {
    res := C._jni_CallObjectMethodA(e.env, C.jobject(obj), C.jmethodID(method), varArgs(args))
    return Object(res), exception(e)
}

func CallBooleanMethod(e Env, obj Object, method MethodID, args ...Value) (bool, error) {
    res := C._jni_CallBooleanMethodA(e.env, C.jobject(obj), C.jmethodID(method), varArgs(args))
    return res == C.JNI_TRUE, exception(e)
}

func CallIntMethod(e Env, obj Object, method MethodID, args ...Value) (int32, error) {
    res := C._jni_CallIntMethodA(e.env, C.jobject(obj), C.jmethodID(method), varArgs(args))
    return int32(res), exception(e)
}

```



```
func main() {  
    a := &App{  
        jvm:      jni.JVMFor(app.JavaVM()),  
        appCtx:   jni.Object(app.AppContext()),  
        updates:  make(chan struct{}, 1),  
        vpnClosed: make(chan struct{}, 1),  
    }  
    a.store = newStateStore(a.jvm, a.appCtx)  
    go func() {  
        if err := a.runBackend(); err != nil {  
            fatalErr(err)  
        }  
    }()  
    go func() {  
        if err := a.runUI(); err != nil {  
            fatalErr(err)  
        }  
    }()  
    app.Main()  
}
```

```
tailscale/ipn/App.java: static native void onSignin(String idToken);
tailscale/ipn/App.java: static native void onVPNPrepared();
tailscale/ipn/App.java: private static native void onConnectivityChanged(boolean connected);
tailscale/ipn/IPNService.java: private native void connect();
tailscale/ipn/IPNService.java: private native void disconnect();
tailscale/ipn/QuickToggleService.java: private static native void onTileClick();
```

```
//export Java_com_tailscale_ipn_App_onSignin
func Java_com_tailscale_ipn_App_onSignin(env *C.JNIEnv, class C.jclass, idToken C.jstring) {
    jenv := jni.EnvFor(uintptr(unsafe.Pointer(env)))
    tok := jni.GoString(jenv, jni.String(idToken))
    onGoogleToken <- tok
}

//export Java_com_tailscale_ipn_IPNService_connect
func Java_com_tailscale_ipn_IPNService_connect(env *C.JNIEnv, this C.jobject) {
    jenv := jni.EnvFor(uintptr(unsafe.Pointer(env)))
    onConnect <- jni.NewGlobalRef(jenv, jni.Object(this))
}

//export Java_com_tailscale_ipn_IPNService_disconnect
func Java_com_tailscale_ipn_IPNService_disconnect(env *C.JNIEnv, this C.jobject) {
    jenv := jni.EnvFor(uintptr(unsafe.Pointer(env)))
    onDisconnect <- jni.NewGlobalRef(jenv, jni.Object(this))
}

//export Java_com_tailscale_ipn_App_onConnectivityChanged
func Java_com_tailscale_ipn_App_onConnectivityChanged(env *C.JNIEnv, cls C.jclass, newConnected C.jboolean) {
    connected.Store(newConnected == C.JNI_TRUE)
    select {
    case onConnectivityChange <- struct{}{}:
    default:
    }
}
```

Synology, QNAP, OPNsense,
FreeBSD, Ubiquiti EdgeRouter, ...

GOOS=freebsd

GOOS=linux GOARCH=mips64le

etc

Server-side: all Go

Database: etcd, also Go

And Tailscale's web admin panel?

TypeScript




```
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less depend
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgr
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/314
npm WARN deprecated request-promise-native@1.0.9: request-promise-native has been deprecated because it extends th
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
```

```
> deasync@0.1.21 install /var/www/tailscale.com/node_modules/deasync
> node ./build.js
```

```
`linux-x64-node-13` exists; testing
Binary is fine; exiting
```

```
> core-js@2.6.12 postinstall /var/www/tailscale.com/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"
```

Thank you for using core-js (<https://github.com/zloirock/core-js>) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:

```
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock
```

Also, the author of core-js (<https://github.com/zloirock>) is looking for a good job -)

```
> parcel-bundler@1.12.4 postinstall /var/www/tailscale.com/node_modules/parcel-bundler
> node -e "console.log('\u001b[35m\u001b[1mLove Parcel? You can now donate to our open collective:\u001b[22m\u001b[0m
```

Love Parcel? You can now donate to our open collective:

```
> https://opencollective.com/parcel/donate
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^2.1.2 (node_modules/jest-haste-map/node_modules/fsevents)
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.1: wanted {"os":"darwin","arch
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.7 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arc
```

added 1229 packages from 616 contributors and audited 1231 packages in 77.675s

```
55 packages are looking for funding
  run `npm fund` for details
```

```
found 1 high severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
```

TypeScript

I like TypeScript.



github.com/evanw/esbuild ?



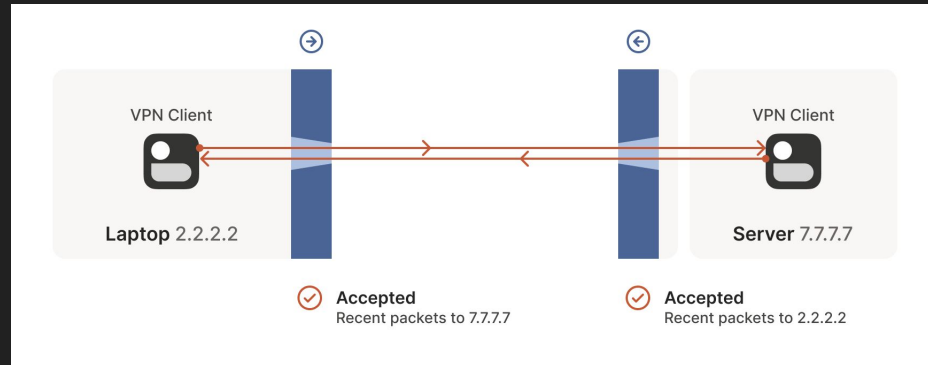
Go WebAssembly something
something?



Adding NAT traversal to WireGuard

How NAT traversal works

tailscale.com/blog/how-nat-traversal-works



golang.zx2c4.com/wireguard

```

// Package conn implements WireGuard's network connections.
package conn

import (
    "errors"
    "net"
    "strings"
)

// A Bind listens on a port for both IPv6 and IPv4 UDP traffic.
//
// A Bind interface may also be a PeekLookAtSocketFd or BindSocketToInterface,
// depending on the platform-specific implementation.
type Bind interface {
    //...
    // ReceiveIPv4 reads an IPv4 UDP packet into b.
    //
    // It reports the number of bytes read, n,
    // the packet source address ep,
    // and any error.
    ReceiveIPv4(b []byte) (n int, ep Endpoint, err error)
    ReceiveIPv6(b []byte) (n int, ep Endpoint, err error)

    // Send writes a packet b to address ep.
    Send(b []byte, ep Endpoint) error

    // Close closes the Bind connection.
    Close() error
}

```


tailscale.com/wgengine/magicsock

type Conn

```
func NewConn(opts Options) (*Conn, error)
func (c *Conn) Close() error
func (c *Conn) CreateBind(uint16) (conn.Bind, uint16, error)
func (c *Conn) CreateEndpoint(pubKey [32]byte, addrs string) (conn.Endpoint, error)
func (c *Conn) DERPs() int
func (c *Conn) DiscoPublicKey() tailcfg.DiscoKey
func (c *Conn) LastMark() uint32
func (c *Conn) LocalPort() uint16
func (c *Conn) PeerHasDiscoKey(k tailcfg.NodeKey) bool
func (c *Conn) Ping(ip netaddr.IP, cb func(*ipnstate.PingResult))
func (c *Conn) ReSTUN(why string)
func (c *Conn) Rebind()
func (c *Conn) ReceiveIPv4(b []byte) (n int, ep conn.Endpoint, addr *net.UDPAddr, err error)
func (c *Conn) ReceiveIPv6(b []byte) (int, conn.Endpoint, *net.UDPAddr, error)
func (c *Conn) Send(b []byte, ep conn.Endpoint) error
func (c *Conn) SetDERPMap(dm *tailcfg.DERPMap)
func (c *Conn) SetMark(value uint32) error
func (c *Conn) SetNetInfoCallback(fn func(*tailcfg.NetInfo))
func (c *Conn) SetNetworkMap(nm *controlclient.NetworkMap)
func (c *Conn) SetNetworkUp(up bool)
func (c *Conn) SetPrivateKey(privateKey wgcfg.PrivateKey) error
func (c *Conn) Start()
func (c *Conn) UpdatePeers(newPeers map[key.Public]struct{})
func (c *Conn) UpdateStatus(sb *ipnstate.StatusBuilder)
```

```
func (c *Conn) ReceiveIPv6(b []byte) (int, conn.Endpoint, *net.L  
func (c *Conn) Send(b []byte, ep conn.Endpoint) error  
func (c *Conn) SetDERPMap(dm *tailcfg.DERPMap)  
func (c *Conn) SetMark(value uint32) error  
func (c *Conn) SetNetInfoCallback(fn func(*tailcfg.NetInfo))  
func (c *Conn) SetNetworkMap(nm *controlclient.NetworkMap)
```

net.IP

```
package net  
type IP []byte
```

mutable

transparent

underlying type is part of the go1 contract,
can't fix after the fact like time.Time

not *comparable*^{*}_—

(can't do == or be a map key)

unnecessarily big

24 byte slice header + separate 4 or 16
byte address array = 28-40 bytes

IPv6 address only 16 bytes

net.IP vs net.IPAddr

a net.IP can't store an IPv6 zone
net.IPAddr does that
(two IP types in std!)

```
bytes.Equal(net.ParseIP("1.2.3.4"),  
            net.ParseIP("::ffff:0102:0304"))
```

inet.af/netaddr.IP

```
package netaddr
```

```
type IP struct {  
    addr      uint128  
    famOrZone *intern.Value  
}  
type uint128 struct {  
    hi, lo uint64  
}
```

- + *immutable*
- + *opaque*
- + *comparable*
- + *24 bytes*
- + *IPv4 != IPv6*
- + *does v6 zones*

go4.org/intern

type Value

```
type Value struct {  
    // contains filtered or unexported fields  
}
```

A Value pointer is the handle to an underlying comparable value. See func Get for how Value pointers may be used.

func (*Value) Get

```
func (v *Value) Get() interface{}
```

Get returns the comparable value passed to
returned v.

func Get ¶

func GetByString

```
func GetByString(s string) *Value
```

```
func Get(cmpVal interface{}) *Value
```

Get returns a pointer representing the comparable value cmpVal.

The returned pointer will be the same for Get(v) and Get(v2) if and only if v == v2, and can be used as a map key.

```
var (  
    // mu guards valMap, a weakref map of *Value by underlying value.  
    // It also guards the resurrected field of all *Values.  
    mu      sync.Mutex  
    valMap  = map[key]uintptr{} // to uintptr(*Value)
```



```
// We play unsafe games that violate Go's rules (and assume a non-moving
// collector). So we quiet Go here.
// See the comment below Get for more implementation details.
//go:nocheckptr
func get(k key) *Value {
    mu.Lock()
    defer mu.Unlock()

    var v *Value
    if valSafe != nil {
        v = valSafe[k]
    } else if addr, ok := valMap[k]; ok {
        v = (*Value)((unsafe.Pointer)(addr))
        v.resurrected = true
    }
    if v != nil {
        return v
    }
    v = k.Value()
    if valSafe != nil {
        valSafe[k] = v
    } else {
        // SetFinalizer before uintptr conversion (theoretical concern;
        // see https://github.com/go4org/intern/issues/13)
        runtime.SetFinalizer(v, finalize)
        valMap[k] = uintptr(unsafe.Pointer(v))
    }
    return v
}
```

```
func finalize(v *Value) {  
    mu.Lock()  
    defer mu.Unlock()  
    if v.resurrected {  
        // We lost the race. Somebody resurrected it while we  
        // were about to finalize it. Try again next round.  
        v.resurrected = false  
        runtime.SetFinalizer(v, finalize)  
        return  
    }  
    delete(valMap, keyFor(v.cmpVal))  
}
```

```
package netaddr
```

```
type IP struct {  
    addr          uint128  
    famOrZone *intern.Value  
}
```

```
// z0, z4, and z6noz are sentinel IP.z values.  
// See the IP type's field docs.  
var (  
    z0      = (*intern.Value)(nil)  
    z4      = new(intern.Value)  
    z6noz   = new(intern.Value)  
)
```

```
// WithZone returns an IP that's the same as ip but with the provided
// zone. If zone is empty, the zone is removed. If ip is an IPv4
// address it's returned unchanged.
```

```
func (ip IP) WithZone(zone string) IP {
    if !ip.Is6() {
        return ip
    }
    if zone == "" {
        ip.z = z6noz
        return ip
    }
    ip.z = intern.GetByString(zone)
    return ip
}
```

```
// Zone returns ip's IPv6 scoped addressing zone, if any.  
func (ip IP) Zone() string {  
    if ip.z == nil {  
        return ""  
    }  
    zone, _ := ip.z.Get().(string)  
    return zone  
}
```

Questions?

Try Tailscale? It's fun.