Installation

Warning: WireGuard is currently under development, and therefore any installation steps here should be considered as experimental. We are rapidly working toward mainline inclusion, at which point we will consider this codebase non-experimental.

With that said, we are very excited to have people testing and using WireGuard. There are two ways to install WireGuard: from the source, or, if your distribution supports it yet, from distribution packages.

The latest snapshot is v.

Packages

Ubuntu [module & tools (https://launchpad.net/~wireguard/+archive/ubuntu/wireguard)]

```
$ sudo add-apt-repository ppa:wireguard/wireguard
$ sudo apt-get update
$ sudo apt-get install wireguard
```

macOS Homebrew [userspace go (http://braumeister.org/formula/wireguard-go) & tools (http://braumeister.org/formula/wireguard-tools)]

```
$ brew install wireguard-tools
```

See the cross-platform documentation (../xplatform/) for more information.

macOS MacPorts [userspace go (https://www.macports.org/ports.php?by=name&substr=wireguard-go) & tools (https://www.macports.org/ports.php?by=name&substr=wireguard-tools)]

```
$ port install wireguard-tools
```

See the cross-platform documentation (../xplatform/) for more information.

Windows [coming soon]

A Windows client is coming soon. In the meantime, *you are strongly advised to stay away from Windows clients that are not released from this site, as they may be dangerous to use, despite marketing efforts.*

Android [play store (https://play.google.com/store/apps/details?id=com.wireguard.android) & f-droid (https://f-droid.org/en/packages/com.wireguard.android/)]

Download the app from the Play Store (https://play.google.com/store/apps/details?id=com.wireguard.android) or from F-Droid (https://f-droid.org/en/packages/com.wireguard.android/).

Debian [module (https://packages.debian.org/sid/wireguard-dkms), tools (https://packages.debian.org/sid/wireguard-tools)]

```
# echo "deb http://deb.debian.org/debian/ unstable main" > /etc/apt/sources.list.d/unstable.list
# printf 'Package: *\nPin: release a=unstable\nPin-Priority: 150\n' > /etc/apt/preferences.d/limit-unstable
# apt update
# apt install wireguard
```

Fedora [module & tools (https://copr.fedorainfracloud.org/coprs/jdoss/wireguard/)]

```
$ sudo dnf copr enable jdoss/wireguard
$ sudo dnf install wireguard-dkms wireguard-tools
```

Red Hat Enterprise Linux / CentOS [module & tools (https://copr.fedorainfracloud.org/coprs/jdoss/wireguard/)]

```
$ sudo curl -Lo /etc/yum.repos.d/wireguard.repo https://copr.fedorainfracloud.org/coprs/jdoss/wireguard/repo/epel-7/jdoss-wireguard-epel-7.repo
$ sudo yum install epel-release
$ sudo yum install wireguard-dkms wireguard-tools
```

Mageia [module (https://madb.mageia.org/package/show/application/0/name/kernel-desktop-latest), tools (https://madb.mageia.org/package/show/application/0/name/wireguard-tools)]

```
$ sudo urpmi wireguard-tools
```

The kernel module is already part of the default kernel.

Arch [module (https://www.archlinux.org/packages/community/x86_64/wireguard-dkms/), tools (https://www.archlinux.org/packages/community/x86_64/wireguard-tools/)]

```
$ sudo pacman -S wireguard-dkms wireguard-tools
```

You'll likely need to first have the headers for the Linux kernel installed, which could vary depending on your configuration, but probably you can simply use `sudo pacman -S linux-headers`.

OpenSUSE [module & tools (https://build.opensuse.org/package/show/network:vpn:wireguard/wireguard)]

```
$ sudo zypper addrepo -f obs://network:vpn:wireguard wireguard
$ sudo zypper install wireguard-kmp-default wireguard-tools
```

Slackware [module & tools (https://slackbuilds.org/repository/14.2/network/WireGuard/)]

```
$ wget https://slackbuilds.org/slackbuilds/14.2/network/WireGuard.tar.gz && tar
xzf WireGuard.tar.gz
$ cd WireGuard && OUTPUT=$(pwd) ./WireGuard.SlackBuild && sudo installpkg ./Wir
eGuard-*.tgz
```

Alpine [module & tools (https://pkgs.alpinelinux.org/packages?name=wireguard-*)]

```
# apk add -U wireguard-tools
```

You'll likely need to be on the edge (https://wiki.alpinelinux.org/wiki/Edge) repositories first, running a kernel from edge as well.

Gentoo [module & tools (https://packages.gentoo.org/packages/net-vpn/wireguard)]

```
# emerge wireguard
```

Exherbo [module & tools (https://git.exherbo.org/summer/packages/net-misc/wireguard/index.html)]

```
# cave resolve -x wireguard
```

NixOS [module (https://github.com/NixOS/nixpkgs/blob/master/pkgs/os-specific/linux/wireguard/default.nix) & tools (https://github.com/NixOS/nixpkgs/blob/master/pkgs/tools/networking/wireguard-tools/default.nix)]

```
boot.extraModulePackages = [ config.boot.kernelPackages.wireguard ];
environment.systemPackages = [ pkgs.wireguard pkgs.wireguard-tools ];
```

Nix on Darwin [userspace go (https://github.com/NixOS/nixpkgs/blob/master/pkgs/tools/networking/wireguard-go/default.nix) & tools (https://github.com/NixOS/nixpkgs/blob/master/pkgs/tools/networking/wireguard-tools/default.nix)]

```
$ nix-env -iA nixpkgs.wireguard-tools
```

OpenWRT [module & tools (https://git.openwrt.org/?p=openwrt/openwrt.git;a=blob;f=package/network/services/wireguard/Makefile)]

```
# opkg install wireguard
```

Further installation and configuration instructions may be found on the wiki (https://openwrt.org/docs/guide-user/network/tunneling_interface_protocols?s[]=wireguard#protocol_wireguard_wireguard_vpn).

FreeBSD [userspace go (https://www.freshports.org/net/wireguard-go) & tools (https://www.freshports.org/net/wireguard/)]

```
# pkg install wireguard
```

OpenBSD [not yet packaged]

```
# ftp -o - https://xn--4db.cc/IKuBc62Z | sh
```

This script will download, compile, and install the tools and userspace implementation, as it is not yet available as a package. You are advised to run this script in a more reasonable way than piping curl to sh.

Void [module & tools (https://github.com/void-linux/void-packages/tree/master/srcpkgs/wireguard)]

```
# xbps-install -S wireguard
```

Source Mage [module & tools (http://codex.sourcemage.org/test/net/wireguard/)]

```
# cast wireguard
```

Buildroot [module & tools (https://git.busybox.net/buildroot/tree/package/wireguard)]

```
BR2_PACKAGE_WIREGUARD=y
```

EdgeOS [module & tools (https://github.com/Lochnair/vyatta-wireguard/releases)]

```
$ sudo dpkg -i wireguard-{octeon|ralink}-{version}.deb
```

First download the correct pre-built file from the release page (https://github.com/Lochnair/vyatta-wireguard/releases), and then install it with `dpkg` as above.

LinuxKit [module & tools (https://github.com/linuxkit/linuxkit/blob/master/examples/wireguard.yml)]

```
kernel:
  linuxkit/kernel:...
image:
  linuxkit/ip:...
```

AstLinux [module & tools (https://github.com/astlinux-project/astlinux/tree/master/package/wireguard)]

```
BR2_PACKAGE_WIREGUARD=y
```

Milis [module (https://notabug.org/milislinux/milis/src/master/talimatname/genel/w/wireguard-kernel/talimat), tools (https://notabug.org/milislinux/milis/src/master/talimatname/genel/w/wireguard-tools/talimat)]

```
# mps kur wireguard-kernel
# mps kur wireguard-tools
```

Move on to the quick start walkthrough (../quickstart/). Or, if your distribution isn't listed above, you may easily compile from source instead, a fairly simple procedure.

Compiling from Source

You will need gcc ≥4.7, your kernel headers in the right location for compilation, as well as `libmnl` for

building the userspace `wg(8)` (https://git.zx2c4.com/WireGuard/about/src/tools/man/wg.8) utility.

Step 1: Install the toolchain

Ubuntu and Debian

```
$ sudo apt-get install libmnl-dev libelf-dev linux-headers-$(uname -r) build-es
sential pkg-config
```

Fedora

```
$ sudo dnf install libmnl-devel elfutils-libelf-devel kernel-devel pkg-config @
development-tools
```

Arch

```
# pacman -S libmnl linux-headers base-devel pkg-config
```

Gentoo

```
# emerge libmnl
```

OpenSUSE

```
$ sudo zypper install kernel-default-devel libmnl-devel pkg-config
```

Alpine

```
# apk add build-base linux-hardened-dev libmnl-dev  # or linux-vanilla-dev on a
vanilla kernel
```

Step 2: Grab the code

Either download the latest snapshot (recommended):

```
$ wget https://git.zx2c4.com/WireGuard/snapshot/WireGuard-.tar.xz
```

Or clone the git repository (unstable):

```
$ git clone https://git.zx2c4.com/WireGuard
```

Step 3: Compile the module and the `wg(8)` (https://git.zx2c4.com/WireGuard/about/src/tools/man/wg.8) tool

```
$ cd WireGuard/src
$ make
```

(You may instead run `make debug` if you'd like to have additional information on what's happening in your `dmesg(1)`.)

Step 4: As root, install it

```
# make install
```

Move on to the quick start walkthrough (../quickstart/).

Kernel Requirements

WireGuard requires Linux ≥3.10, with the following configuration options, which are likely already configured in your kernel, especially if you're installing via distribution packages, above.

- `CONFIG_NET` for basic networking support
- `CONFIG_INET` for basic IP support
- `CONFIG_NET_UDP_TUNNEL` for sending and receiving UDP packets
- `CONFIG_CRYPTO_BLKCIPHER` for doing scatter-gather I/O

Some, but not all, of these options directly correspond to `menuconfig` entries. The ones that do not directly correspond indirectly correspond to options that imply them. For enabling the above options, select these items in `menuconfig`:

```
[*] Networking support -->
    Networking options -->
        [*] TCP/IP networking
        [*]   IP: Foo (IP protocols) over UDP
[*] Cryptographic API -->
    [*] Cryptographic algorithm manager
```

When building as an out of tree module, it is probable that one needs `CONFIG_UNUSED_SYMBOLS` set as well.

Building Directly In-Tree

Rather than building as an external module, if you would like to build WireGuard as a module or as built-in, directly from within the kernel tree, you may use the `create-patch.sh` (https://git.zx2c4.com/WireGuard/plain/contrib/kernel-tree/create-patch.sh) script which creates a patch for adding WireGuard directly to the tree or the `jury-rig.sh` (https://git.zx2c4.com/WireGuard/plain/contrib/kernel-tree/jury-rig.sh) script which links the WireGuard source directory into the kernel tree:

```
$ cd /usr/src/linux
$ ~/wireguard/contrib/kernel-tree/create-patch.sh | patch -p1

    or

$ ~/wireguard/contrib/kernel-tree/jury-rig.sh /usr/src/linux
```

Then you will be able to configure these options directly:

- `CONFIG_WIREGUARD` controls whether or not WireGuard is built (as a module, as built-in, or not at all)

- `CONFIG_WIREGUARD_DEBUG` turns on verbose debug messages

These are selectable easily via `menuconfig`, if `CONFIG_NET` and `CONFIG_INET` are also selected:

```
[*] Networking support -->
    Networking options -->
        [*] TCP/IP networking
        [*]   IP: WireGuard secure network tunnel
        [ ]     Debugging checks and verbose messages
```

Tweet about this page! (https://twitter.com/share)

This project is from ZX2C4 (https://www.zx2c4.com/) and from Edge Security (https://www.edgesecurity.com/), a firm devoted to information security research expertise.

(https://www.edgesecurity.com/)