NAME

wg - set and retrieve configuration of WireGuard interfaces

SYNOPSIS

wg [*COMMAND*] [*OPTIONS*]... [*ARGS*]...

DESCRIPTION

wg is the configuration utility for getting and setting the configuration of WireGuard tunnel interfaces. The interfaces themselves can be added and removed using **ip-link**(8) and their IP addresses and routing tables can be set using **ip-address**(8) and **ip-route**(8). The **wg** utility provides a series of sub-commands for changing WireGuard-specific aspects of WireGuard interfaces.

If no COMMAND is specified, COMMAND defaults to **show**. Sub-commands that take an INTERFACE must be passed a WireGuard interface.

COMMANDS

show { *<interface>* | *all* | *interfaces* } [*public-key* | *private-key* | *listen-port* | *fwmark* | *peers* | *preshared-keys* | *endpoints* | *allowed-ips* | *latest-handshakes* | *persistent-keepalive* | *transfer* | *dump*]

Shows current WireGuard configuration and runtime information of specified *<interface>*. If no *<interface>* is specified, *<interface>* defaults to *all*. If *interfaces* is specified, prints a list of all WireGuard interfaces, one per line, and quits. If no options are given after the interface specification, then prints a list of all attributes in a visually pleasing way meant for the terminal. Otherwise, prints specified information grouped by newlines and tabs, meant to be used in scripts. For this script-friendly display, if *all* is specified, then the first field for all categories of information is the interface name. If *dump* is specified, then several lines are printed; the first contains in order separated by tab: private-key, public-key, listen-port, fwmark. Subsequent lines are printed for each peer and contain in order separated by tab: public-key, preshared-key, endpoint, allowed-ips, latest-handshake, transfer-rx, transfer-tx, persistent-keepalive.

showconf *<interface>*

Shows the current configuration of *<interface>* in the format described by *CONFIGURATION FILE FORMAT* below.

set *<interface>* [*listen-port <port>*] [*fwmark <fwmark>*] [*private-key <file-path>*] [*peer <base64-public-key>* [*remove*] [*preshared-key <file-path>*] [*endpoint <ip>:<port>*] [*persistent-keepalive <interval seconds>*] [*allowed-ips <ip1>/<cidr1>*[,*<ip2>/<cidr2>*]...]]...

Sets configuration values for the specified *<interface>*. Multiple *peer*s may be specified, and if the *remove* argument is given for a peer, that peer is removed, not configured. If *listen-port* is not specified, or set to 0, the port will be chosen randomly when the interface comes up. Both *private-key* and *preshared-key* must be files, because command line arguments are not considered private on most systems but if you are using **bash**(1), you may safely pass in a string by specifying as *private-key* or *preshared-key* the expression: <(echo PRIVATEKEYSTRING). If */dev/null* or another empty file is specified as the filename for either *private-key* or *preshared-key*, the key is removed from the device. The use of *preshared-key* is optional, and may be omitted; it adds an additional layer of symmetric-key cryptography to be mixed into the already existing public-key cryptography, for post-quantum resistance. If *allowed-ips* is specified, but the value is the empty string, all allowed ips are removed from the peer. The use of *persistent-keepalive* is optional and is by default off; setting it to 0 or "off" disables it. Otherwise it represents, in seconds, between 1 and 65535 inclusive, how often to send an authenticated empty packet to the peer, for the purpose of keeping a stateful firewall or NAT mapping valid persistently. For example, if the interface very rarely sends traffic, but it might at anytime receive traffic from a peer, and it is behind NAT, the interface might benefit from having a persistent keepalive interval of 25 seconds; however, most users will not need this. The use of *fwmark* is optional and is by default off; setting it to 0 or "off" disables it. Otherwise it is a 32-bit fwmark for outgoing packets and may be specified in hexadecimal by prepending "0x".

setconf *<interface> <configuration-filename>*

Sets the current configuration of *<interface>* to the contents of *<configuration-filename>*, which must be in the format described by *CONFIGURATION FILE FORMAT* below.

addconf *<interface> <configuration-filename>*

Appends the contents of *<configuration-filename>*, which must be in the format described by *CONFIGURATION FILE FORMAT* below, to the current configuration of *<interface>*.

syncconf *<interface> <configuration-filename>*

Like **setconf**, but reads back the existing configuration first and only makes changes that are explicitly different between the configuration file and the interface. This is much less efficient than **setconf**, but has the benefit of not disrupting current peer sessions. The contents of *<configuration-filename>* must be in the format described by *CONFIGURATION FILE FORMAT* below.

genkey

Generates a random *private* key in base64 and prints it to standard output.

genpsk

Generates a random *preshared* key in base64 and prints it to standard output.

pubkey

Calculates a *public* key and prints it in base64 to standard output from a corresponding *private* key (generated with *genkey*) given in base64 on standard input.

A private key and a corresponding public key may be generated at once by calling:

```
$ umask 077
$ wg genkey | tee private.key | wg pubkey > public.key
```

help

Shows usage message.

CONFIGURATION FILE FORMAT

The configuration file format is based on *INI*. There are two top level sections -- *Interface* and *Peer*. Multiple *Peer* sections may be specified, but only one *Interface* section may be specified.

The *Interface* section may contain the following fields:

- PrivateKey -- a base64 private key generated by *wg genkey*. Required.

- ListenPort -- a 16-bit port for listening. Optional; if not specified, chosen randomly.

- FwMark -- a 32-bit fwmark for outgoing packets. If set to 0 or "off", this option is disabled. May be specified in hexadecimal by prepending "0x". Optional.

The *Peer* sections may contain the following fields:

- PublicKey -- a base64 public key calculated by *wg pubkey* from a private key, and usually transmitted out of band to the author of the configuration file. Required.

- PresharedKey -- a base64 preshared key generated by *wg genpsk*. Optional, and may be omitted. This option adds an additional layer of symmetric-key cryptography to be mixed into the already existing public-key cryptography, for post-quantum resistance.

- AllowedIPs -- a comma-separated list of IP (v4 or v6) addresses with CIDR masks from which incoming traffic for this peer is allowed and to which outgoing traffic for this peer is directed. The catch-all *0.0.0.0/0* may be specified for matching all IPv4 addresses, and *::/0* may be specified for matching all IPv6 addresses. May be specified multiple times.

- Endpoint -- an endpoint IP or hostname, followed by a colon, and then a port number. This endpoint will be updated automatically to the most recent source IP address and port of correctly authenticated packets from the peer. Optional.

- PersistentKeepalive -- a seconds interval, between 1 and 65535 inclusive, of how often to send an authenticated empty packet to the peer for the purpose of keeping a stateful firewall or NAT mapping valid persistently. For example, if the interface very rarely sends traffic, but it might at anytime receive traffic from a peer, and it is behind NAT, the interface might benefit from having a persistent keepalive interval of 25 seconds. If set to 0 or "off", this option is disabled. By default or when unspecified, this option is off. Most users will not need this. Optional.

CONFIGURATION FILE FORMAT EXAMPLE

This example may be used as a model for writing configuration files, following an INI-like syntax. Characters after and including a ’#’ are considered comments and are thus ignored.

```
[Interface]
PrivateKey = yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=
ListenPort = 51820

[Peer]
PublicKey = xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=
Endpoint = 192.95.5.67:1234
AllowedIPs = 10.192.122.3/32, 10.192.124.1/24

[Peer]
PublicKey = TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=
Endpoint = [2607:5300:60:6b0::c05f:543]:2468
AllowedIPs = 10.192.122.4/32, 192.168.0.0/16

[Peer]
PublicKey = gN65BkIKy1eCE9pP1wdc8ROUtkHLF2PfAqYdyYBz6EA=
Endpoint = test.wireguard.com:18981
AllowedIPs = 10.10.10.230/32
```

DEBUGGING INFORMATION

Sometimes it is useful to have information on the current runtime state of a tunnel. When using the Linux kernel module on a kernel that supports dynamic debugging, debugging information can be written into **dmesg**(1) by running as root:

```
# modprobe wireguard && echo module wireguard +p > /sys/kernel/debug/dynamic_debug/control
```

On OpenBSD and FreeBSD, debugging information can be written into **dmesg**(1) on a per-interface basis by using **ifconfig**(1):

```
# ifconfig wg0 debug
```

On userspace implementations, it is customary to set the *LOG_LEVEL* environment variable to *verbose*.

ENVIRONMENT VARIABLES

WG_COLOR_MODE

If set to *always*, always print ANSI colorized output. If set to *never*, never print ANSI colorized output. If set to *auto*, something invalid, or unset, then print ANSI colorized output only when writing to a TTY.

WG_HIDE_KEYS

If set to *never*, then the pretty-printing **show** sub-command will show private and preshared keys in the output. If set to *always*, something invalid, or unset, then private and preshared keys will be printed as "(hidden)".

WG_ENDPOINT_RESOLUTION_RETRIES

If set to an integer or to *infinity*, DNS resolution for each peer’s endpoint will be retried that many times for non-permanent errors, with an increasing delay between retries. If unset, the default is 15 retries.

SEE ALSO

wg-quick(8), **ip**(8), **ip-link**(8), **ip-address**(8), **ip-route**(8).

AUTHOR

wg was written by Jason A. Donenfeld <Jason@zx2c4.com>. For updates and more information, a project page is available on the World Wide Web <https://www.wireguard.com/>.