Install and configure WireGuard VPN server onto a HamGate

Mark Phillips, NI2O 20221231 V1.14

Table of Contents

Version Control

Author	Notes	Date
Mark Phillips, NI2O	Initial Creation	04/21/2022
Mark Phillips, NI2O	Updates	12/31/2022

1 Assumptions

The following assumptions have been made prior to this install;

Linux (preferably Debian) operating system
Correctly functioning IP routing layer

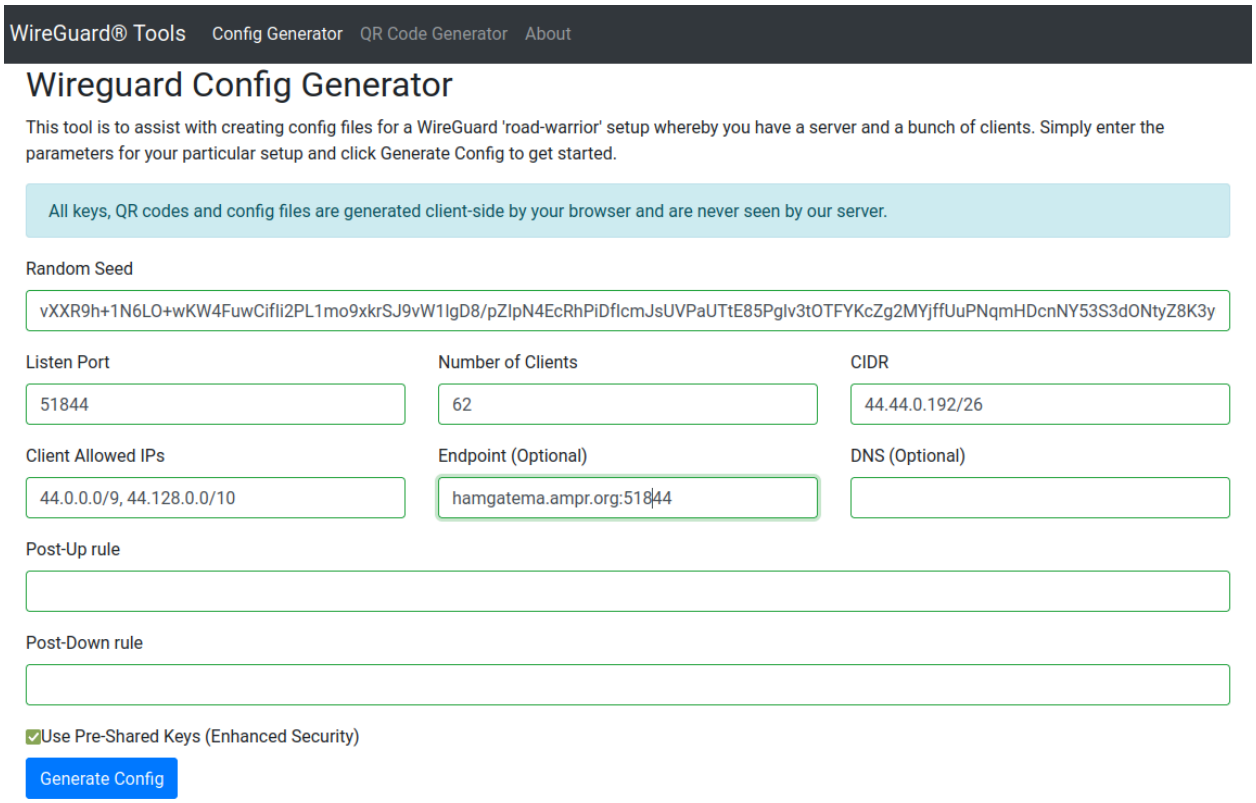
2 Required software

Install the Wireguard VPN software for your distribution;

Debian: apt install wireguard-dkms
RedHat: yum install wireguard

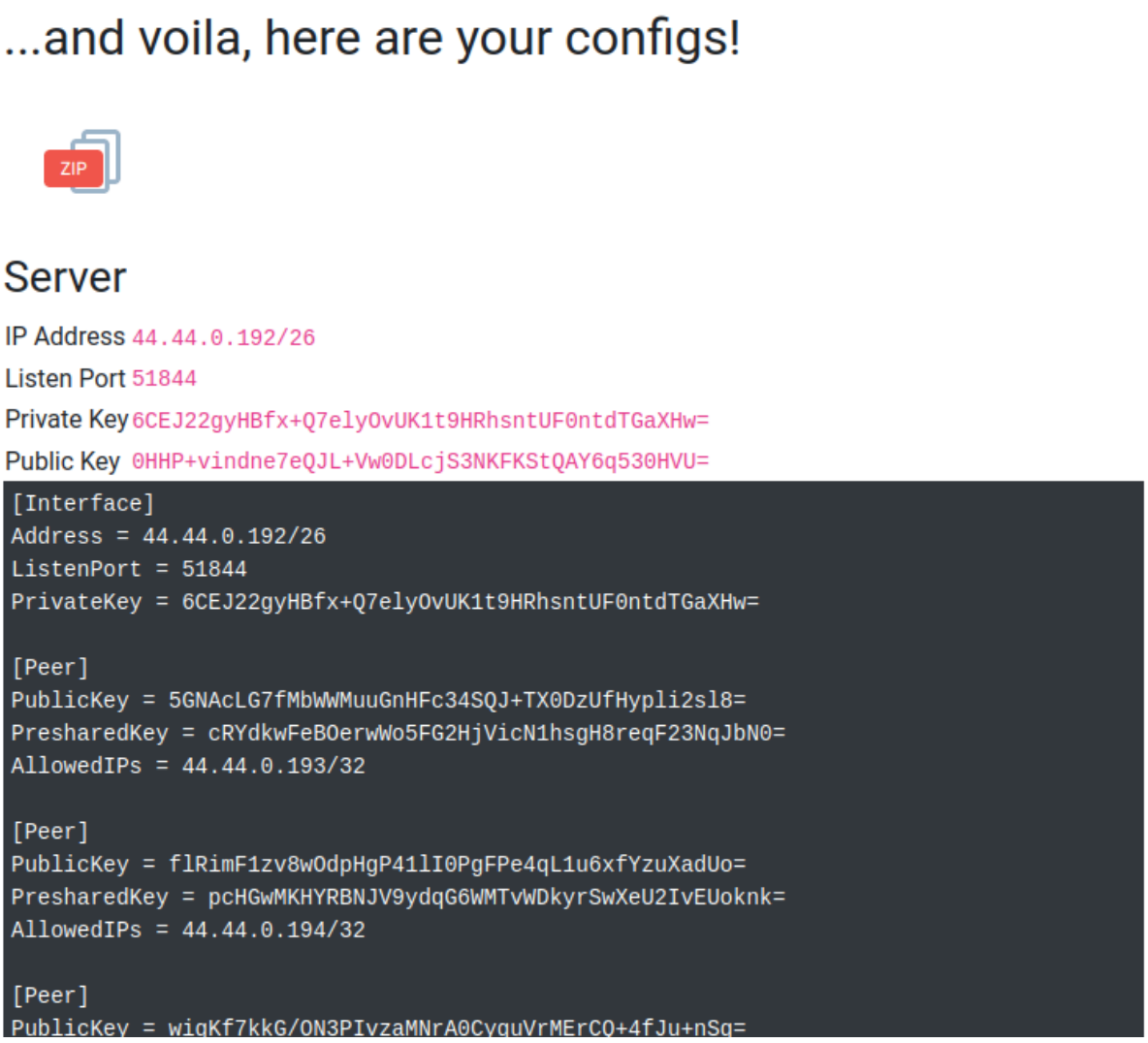
3 Create configs

The simplest way to create the initial configuration files is to use the WireGuard Config Generator tool found at https://www.wireguardconfig.com/ Create your initial config as in the example below. Ensure to remove the pre-populated Post-UP and Post-Down rules.



Ensure that the “Client Allowed IPs” field contains ONLY the 44Net networks you wish to support. 44Net is not in the ISP business!

Further down the page you'll see a printout of your configurations and a link to download them as a .zip file. There is one config file for the server and additional config files for each client. These files are all unique. Each client will require a unique config file.



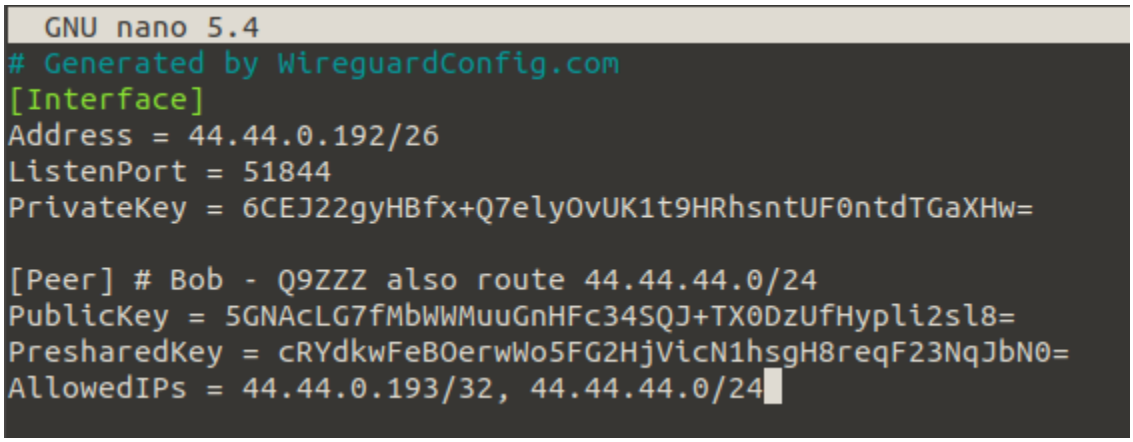
Transfer the .zip file to your HamGate placing it in the /etc/wireguard directory.

Unzip the config file into the /etc/wireguard directory. A subdirectory called "configs" will be created which contains all of the required files. Copy the server.conf file from /etc/wireguard/configs to /etc/wireguard. You may rename it to something useful if required. NB: the name of the config file becomes the name of the network interface created by the VPN server.

4 Adjust configs

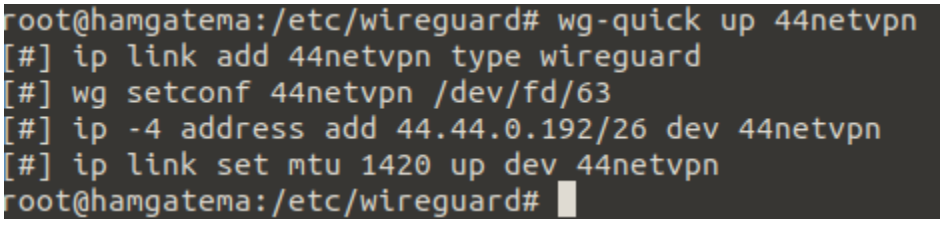
Ensure that the VPN is not running by doing ‘wg-quick down [server config filename]’. This must be executed before every adjustment. If you do not do this, a restart of the VPN will create errors as changes will not be removed before they are inserted.

It may be necessary to make adjustments to the config files created by the config tool. As created, the configs only allow the server to talk to a given IP address issued to a client. If your client has (insert pronoun here) own subnet that they wish to connect to the VPN then an allowance must be made for this in the server config file. While you’re at it make a few notes in the file so that you know what belongs to whom. A demonstration is shown below.



5 Test configs

Run the VPN by executing ‘wg-quick up [server config filename]’. If all went well a few lines of config will appear as WireGuard makes the necessary adjustments to your routing table and creates the VPN interface.



To check the VPN is running do an ‘ifconfig’ and look for an interface with the label of your config file (44netvpn in our example).

```
root@hamgatema:/etc/wireguard# ifconfig
44netvpn: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1420
        inet 44.44.0.192  netmask 255.255.255.192  destination 44.44.0.192
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

On your test client, copy a config file from the /etc/wireguard/configs directory on your server to /etc/wireguard on your client machine.

```
 GNU nano 4.8
# Generated by WireguardConfig.com
[Interface]
Address = 44.44.0.254/26
ListenPort = 51844
PrivateKey = IIIUxWdWBu2ETDteXHiN7valFcLNPqjRsmWsEOQOt2A=

[Peer]
PublicKey = 0HHP+vindne7eQJL+Vw0DLcjS3NKFKStQAY6q530HVU=
PresharedKey = 9cM55VFlDKPGcwkA/t0WIJj6TVgACPQLN+u6hP5h4ag=
AllowedIPs = 44.0.0.0/9, 44.128.0.0/10
Endpoint = hamgatema.ampr.org:51844
```

Start the client in the same way as you did on the server with ‘wg quick up [client config filename]’. You should see some lines telling you that the VPN is installing routes etc.

```
g7ltt@mphillips-ubuntu:~$ sudo wg-quick up ./hamgatema.conf
Warning: `/home/g7ltt/hamgatema.conf' is world accessible
[#] ip link add hamgatema type wireguard
[#] wg setconf hamgatema /dev/fd/63
[#] ip -4 address add 44.44.0.254/26 dev hamgatema
[#] ip link set mtu 1420 up dev hamgatema
[#] ip -4 route add 44.128.0.0/10 dev hamgatema
[#] ip -4 route add 44.0.0.0/9 dev hamgatema
g7ltt@mphillips-ubuntu:~$
```

Confirm the link has been established by ‘wg show’.

```
g7ltt@mphillips-ubuntu:~$ sudo wg show
interface: hamgatema
  public key: ZSXCylrD6feP+oW+l0rH/jpPxjoliQT043BFgdvBRgQ=
  private key: (hidden)
  listening port: 51844

peer: 0HHP+vindne7eQJL+Vw0DLcjS3NKFKStQAY6q530HVU=
  preshared key: (hidden)
  endpoint: 45.79.3.86:51844
  allowed ips: 44.0.0.0/9, 44.128.0.0/10
  latest handshake: 1 minute, 3 seconds ago
  transfer: 92 B received, 244 B sent
  persistent keepalive: every 25 seconds
g7ltt@mphillips-ubuntu:~$
```

Now traceroute to a device you know to be on the 44Net. This demonstrates that not only does the VPN link work but that you are traversing the 44Net rather than the public Internet.

```
g7ltt@mphillips-ubuntu:~$ traceroute 44.4.50.2
traceroute to 44.4.50.2 (44.4.50.2), 30 hops max, 60 byte packets
 1  * * *
 2  w2xsc-gw.ampr.org (44.4.50.10)  96.260 ms  96.209 ms  98.072 ms
 3  w2xsc-tj.ampr.org (44.4.50.21)  101.934 ms  101.431 ms  104.828 ms
g7ltt@mphillips-ubuntu:~$
```

The first hop of the traceroute will not resolve an IP address as this is the VPN server.

6 Autorun VPN at boot

It would be a good idea to bring up the VPN every time the host server is rebooted. Follow the below howto for further details.
https://www.ivpn.net/knowledgebase/linux/linux-autostart-wireguard-in-systemd/

7 Overcoming NAT issues

NAT: it's a fantastic solution while at the same time it's a PITA! The problem is this; A client is behind a NAT router (eg cable modem at home). Every time a connection is made out of their network a timer is started on the connection by their firewall/router. If there is no VPN traffic to send WireGuard will not transmit anything and so the firewall/router will close the connection after the timer expires.

This can be overcome by forcing the client to keep the connection open with a “keepalive”. If keepalives are required edit the client config file as in the example below by adding the PersistantKeepalive statement. A value of 25 seconds is usually enough.

```
  GNU nano 4.8
# Generated by WireguardConfig.com
[Interface]
Address = 44.44.0.254/26
ListenPort = 51844
PrivateKey = IIIUxWdWBu2ETDteXHiN7valFcLNPqjRsmWsEOQOt2A=

[Peer]
PublicKey = 0HHP+vindne7eQJL+Vw0DLcjS3NKFKStQAY6q530HVU=
PresharedKey = 9cM55VFlDKPGcwkA/t0WIJj6TVgACPQLN+u6hP5h4ag=
AllowedIPs = 44.0.0.0/9, 44.128.0.0/10
Endpoint = hamgatema.ampr.org:51844
PersistentKeepalive = 25
```