ChatGPT

Using Fedora 42 + WireGuard to Route Free.fr Services via a French IP

Problem Overview: *Free.fr* (a French ISP) restricts certain personal hosting services (FTP, stats, etc.) to connections originating from within France. For example, Free's FTP for personal pages (`ftpperso.free.fr`) **blocks logins from non-French IP addresses**, requiring a VPN or proxy if you're outside France [1]. We have a Fedora 42 server acting as a local network router (with DNS/DHCP via `dnsmasq` and multiple interfaces) and want to leverage it – along with WireGuard VPN – to ensure that any FTP/HTTP requests to Free's personal site domains use a **Metropolitan France IP address**. The target Free.fr resources include:

- `ftpperso.free.fr` – FTP host for uploading personal websites
- `free.fr` – Free's main site (used for account auth/management)
- `statsperso.free.fr` – personal site statistics interface
- `st.free.fr` – Free's "phpStats" subdomain for stats [2]

Our goal is to configure the Fedora server as a **VPN exit node** with a French IP for traffic to these domains. Below we outline how to do this using WireGuard, and discuss alternative solutions for automation and multi-server setups.

1. Environment Setup & Prerequisites

Existing Fedora Router Setup: The Fedora 42 server is already configured as a network gateway for our LAN and lab "exosystem." It provides DHCP and DNS (via NetworkManager + dnsmasq) and routes traffic from internal interfaces to the WAN. Key baseline configurations include enabling IP forwarding and NAT masquerading on the WAN interface [3]. For example, the setup script enables kernel IP forwarding and installs a firewall rule to NAT (masquerade) outbound traffic from the LAN or VPN through the WAN (internet) interface [4] [3]. This means any device on the LAN or any WireGuard peer in the internal network can have its traffic forwarded out to the internet via Fedora's public interface.

WireGuard Installed: WireGuard is installed and an initial VPN (`wg0`) is configured for internal connectivity. In our setup, `wg0` serves as a LAN/VPN bridge (peers get 10.44.x.x addresses, etc.). We will add a **new WireGuard configuration** (or peer) specifically to obtain a French egress IP. You'll need:

- **A French VPN Endpoint:** This can be a **self-hosted server in France** (e.g. a VPS in a French datacenter or a machine on a French ISP) or a **VPN service** that offers WireGuard endpoints in France. For instance, you might rent a small VM in Paris, or use a VPN provider's WireGuard config for France. Ensure this endpoint has a French public IP and is set up to forward traffic.

- *If self-hosting:* Install WireGuard on the French server and configure it to accept a peer (the Fedora box) with AllowedIPs covering the peer's tunnel IP *and* any traffic the peer will send (we'll use

0.0.0.0/0 for full tunnel, or specific subnets for split tunnel). Enable IP forwarding and NAT on that French server so it can act as an exit node (similar to how Fedora is set up, but in France).

- *If using a provider:* Obtain the WireGuard client configuration for a French exit. Providers often supply a config with `AllowedIPs = 0.0.0.0/0` (meaning the VPN will be used as the default route).

- **Credentials/Keys:** Generate WireGuard keypairs for the Fedora client and (if self-hosted) the French server. Exchange public keys and endpoints accordingly.

2. WireGuard Configuration for French Exit

On the Fedora server, create a new WireGuard interface (e.g. `wg-fr.conf`) for the connection to the French node. This will give Fedora a **virtual interface that tunnels out via a French IP**. Key points in the config:

- **Interface Address:** Assign an IP from a private range for the tunnel (must match the range used on the French side). For example, if the French server's WireGuard interface is 10.200.0.1/24, give Fedora `10.200.0.2/24`. This is just for the tunnel's internal routing.

- **Peer (French server) settings:** Include the French server's public key and endpoint (IPv4 or DNS + port).

- **Allowed IPs:** This is crucial – it controls what traffic goes into the VPN tunnel. To route *only the Free.fr service traffic* via France, you can specify the IP ranges corresponding to those domains. For example, Free's pages and FTP servers reside in Free's network (e.g. `ftpperso.free.fr` resolves to an IP in the `212.27.63.x` range [5]). You might set:

```
AllowedIPs = 212.27.0.0/16, 213.228.0.0/16
```

This would cover a broad swath of Free's IPv4 addresses (Free SAS owns multiple ranges in 212.x and 213.x). You can tighten this to specific subnets if known (for instance, personal pages FTP and web might be in `212.27.60.0/24` or `212.27.63.0/24`). The idea is to include any IPs for `free.fr`, `ftpperso.free.fr`, `statsperso.free.fr`, and `st.free.fr` so that **any traffic to those addresses will be routed via the WireGuard tunnel**.

Tip: If unsure of all subnets, using a larger netblock or even `0.0.0.0/0` is simpler and guarantees the French exit will be used for all internet traffic from Fedora when the tunnel is up. Setting `AllowedIPs = 0.0.0.0/0` on the Fedora peer config tells WireGuard to route **all egress traffic** over the tunnel [6]. This "full tunnel" mode ensures the French IP is used for everything (you can then selectively use it only when needed). For more fine-grained control, use the specific ranges so only Free-related traffic goes through VPN and the rest uses your normal link.

- **Routing/NAT on Fedora:** When the `wg-fr` interface is up, Fedora will add routes for the Allowed IPs via the WireGuard interface. For example, if AllowedIPs includes `212.27.0.0/16`, Fedora's routing table will direct that subnet to the `wg-fr` interface. Because Fedora already has IP

forwarding and masquerade enabled, any LAN clients or processes on Fedora itself using those IPs will automatically be forwarded through the tunnel. The Fedora server will NAT their source to the WireGuard tunnel IP (or ultimately to the French server's IP when exiting). The return traffic will come back through the French server and tunnel, and Fedora will forward it back to the client.

No extra firewall rules should be needed beyond what's already in place – since our Fedora's firewall (firewalld/iptables) is set to masquerade outbound traffic on the WAN, it should similarly NAT outbound traffic on the VPN (the WireGuard interface was added to the internal zone in our setup [7], so it's treated like LAN – allowed to forward – and then masqueraded out the WAN). The French exit server will also do NAT to its own WAN if configured as typical. In short, once the tunnel is up, it behaves like a secured "gateway" for those destinations.

- **DNS Consideration:** Ensure Fedora's DNS (dnsmasq) can resolve those Free.fr domains as usual. The DNS queries themselves don't need to go via France; only the actual FTP/HTTP connections do. (Free's services don't appear to geo-restrict DNS, just the actual service access.) If paranoid or for completeness, you could use a French DNS server for those queries, but it's typically unnecessary. The connections will be made to the correct IPs and then routed appropriately.

Example WireGuard client config (Fedora side) – assuming a self-hosted scenario:

```
# /etc/wireguard/wg-fr.conf (Fedora client)
[Interface]
PrivateKey = <FedoraPrivateKey>
Address = 10.200.0.2/24
# DNS = 10.200.0.1  (optional, if you want to use remote DNS)

[Peer]
PublicKey = <FrenchServerPublicKey>
Endpoint = <FrenchServerPublicIP>:51820
# Route Free.fr services via this peer (use specific ranges or 0.0.0.0/0 for
full tunnel)
AllowedIPs = 212.27.0.0/16, 213.228.0.0/16, 212.Free.others/??
PersistentKeepalive = 25   # keepalive to maintain NAT traversal, if needed
```

And on the French server's `/etc/wireguard/wg0.conf`, the Fedora peer might be defined like:

```
[Peer] # Fedora42 Router
PublicKey = <FedoraPublicKey>
AllowedIPs = 10.200.0.2/32
```

(The French server can also set `AllowedIPs = 10.200.0.2/32, 192.168.1.0/24` etc. if you want it to know about your LAN, but that's optional for our use-case. It mainly needs to accept the Fedora's tunnel IP.) On the French side, don't forget to enable IP forwarding and add a MASQUERADE rule on its outbound interface, so it can act as a proper VPN gateway, allowing Fedora's traffic to exit with the French IP.

Bring up the tunnel: Use `wg-quick up wg-fr` (or NetworkManager if you integrate it as a connection profile) on Fedora. The WireGuard interface should come up and the routes be added. You can verify by running `ip route` – you should see the Free.fr network ranges pointing to `wg-fr`. Also `wg show` should indicate the tunnel is established.

Testing: Now test an FTP/HTTP connection to Free's services from a client behind Fedora or from Fedora itself:

- For FTP (ftpperso): Try connecting with your credentials. It should succeed now if the tunnel is working, whereas it would timeout or refuse before from a non-French IP. The traffic will exit via the French VPN, satisfying Free's IP check. *Note:* FTP opens separate data channels for listings and transfers, but because **we route all traffic to the FTP server's IP through the VPN**, those data connections will also go via France. (If we attempted policy routing by port only, it would fail for FTP's additional connections [8], so routing by IP address ensures all FTP control/data traffic consistently uses the VPN path.)

- For web (statsperso.free.fr or free.fr): You can use `curl http://statsperso.free.fr/...` or simply observe via a web browser from a LAN PC. The external request should be going out via France. Checking a service like `ifconfig.me` from the Fedora machine while the tunnel is up could show the French IP if you temporarily route all traffic, but if split by IP, just testing the actual services is enough.

At this point, **any authentication, hosting upload, or stats retrieval actions involving those Free.fr domains will utilize the French IP** provided by the WireGuard tunnel. The Fedora box essentially becomes an "exit node" for those specific services.

3. Automating and Integrating into Workflows

Beyond the manual setup, you likely want this to work seamlessly as part of automated deployments or distributed systems (Fedora/Ubuntu servers in an "interweb" architecture). Here are some tips for automation and alternative implementations:

- **Persisting & Auto-Starting VPN:** Since this Fedora router runs continuously, you can enable the WireGuard client to auto-start on boot. For example, if using `wg-quick`, enable the systemd service `wg-quick@wg-fr.service`. In NetworkManager, you could create a WireGuard type connection and set `autoconnect=true`. This ensures the VPN comes up whenever the system reboots or network reinitializes, maintaining the French link available for any scripts that need it.

- **Integration in Deployment Scripts:** If you have deployment processes (CI/CD pipelines or cron jobs) that publish content to Free.fr, integrate the VPN bring-up and tear-down in those scripts. For instance, an update script could do:

```
nmcli connection up wg-fr  # or wg-quick up wg-fr
lftp -u user,pass ftpperso.free.fr -e "mirror -R site/ / ; quit"
nmcli connection down wg-fr  # bring down if you don't need it always on
```

This way, the VPN is only up during the upload. However, given the minimal overhead of WireGuard, you might just keep it up persistently and only route traffic when needed. The overhead is low, and using AllowedIPs means it's not affecting other traffic except the specified domains.

- **Monitoring and Failover:** You can monitor the tunnel's health by pinging the French endpoint or using WireGuard's built-in keepalive. If the French VPN server goes down, you might lose access to Free's services again. It could be wise to have a fallback (another server or even use an alternative method like SSH tunnel) in case of VPN outage. Tools like cron or systemd can periodically check connectivity (for example, attempt to `ftp ftpperso.free.fr` and alert if it fails) so you know if the tunnel is not functioning.

- **Multiple Servers (Ubuntu, etc.):** If you have other servers (Ubuntu or Fedora) distributed elsewhere that also need to access Free.fr services, you have two main options:

- **Give each server its own French VPN connection:** Install WireGuard on each and either connect each to the same French exit server (using unique keys and tunnel IPs per server), or use a VPN provider's config on each. This is straightforward and keeps each server independent.

- **Route via the central Fedora router:** Since we already have a WireGuard mesh/lan (the "exosystem"), other machines could be WireGuard peers to the Fedora router (as some `wg0` peers are configured [9]). In that case, an Ubuntu server peer could simply send its Free.fr traffic through the Fedora router. Fedora is already doing NAT for its WireGuard peers over its WAN [3] – and once Fedora itself uses the French exit for Free.fr, the Ubuntu peer's traffic destined for those IPs will also egress in France. This is somewhat complex (traffic goes Ubuntu -> Fedora via wg0, then Fedora -> French exit via wg-fr). If latency or double encryption is not an issue, this centralization can simplify having only one egress point. Just ensure the Ubuntu peer's AllowedIPs on its connection to Fedora includes the Free.fr IPs or default route so that it knows to send that traffic to Fedora. In practice, it might be easier to just give each server its own direct tunnel to France unless you have a strong reason to funnel through one hub.

- **Alternative VPN Technologies:** While WireGuard is lightweight and easy to automate (with simple config files or even QR codes for peers), you could accomplish the same with other VPNs:

- *OpenVPN/IPsec:* An OpenVPN client on Fedora (or on each server) connecting to a French server can tunnel the traffic similarly. This might be useful if you already have an OpenVPN server in France or need features like username/password auth. The principle (split-tunnel routes or full tunnel) is the same, but configuration will differ (you'd use `client config` with `route 212.27.0.0 255.255.0.0` for example).

- *SSH Tunneling/Proxy:* In a pinch, an SSH tunnel can forward traffic too. For web HTTP requests, one could use an SSH dynamic SOCKS proxy on a French host (`ssh -D1080 user@french-host`) and then configure tools to use that proxy. For FTP, one could use SSH local port forwarding for the command channel and a proxy command for data, but FTP is hard to fully support via simple port forwards due to its multi-connection nature. There are FTP-specific proxy solutions (as Free's note suggests, using a "proxy FTP" with a French IP) [1], but those are less common. If you only need to transfer files occasionally, an easier alternative is to **run the FTP client on the French server** (or a French VM) directly – for example, use `scp` to send files to the French box, then locally ftp from

there to Free. This can be scripted but introduces an extra hop and complexity (and storing credentials on the remote).

- *Web-based upload:* Free.fr provides a web interface for uploads (called "WebFTP") accessible through your account on free.fr. This is a manual method (or one could script a headless browser to do it), but it's not really meant for automation. It's mentioned here as Free's suggested workaround for users abroad (along with VPN) [1].

- **Cloud Automation:** If you prefer not keeping a permanent server in France, you could dynamically create an exit node when needed. For example, using tools like Terraform and cloud APIs to spin up a transient VM in France, configure WireGuard on it via cloud-init, connect your server, do the transfers, then tear it down. While overkill for most, it's an option for a fully automated, on-demand approach (similar techniques are used in some advanced devops setups [10] [11]).

In summary, the **WireGuard solution** turns your Fedora router into a service that ensures all interactions with Free.fr's personal site platforms are coming from a French IP. This satisfies Free's requirements and allows automated deployment and management tasks (like publishing your website or fetching stats) to run smoothly from anywhere. We used WireGuard for its simplicity and performance, but other VPN or proxy methods can achieve the same goal. The key steps are: obtain a French egress point, route the necessary traffic through it (e.g., by configuring AllowedIPs or routes for Free.fr addresses), and integrate the VPN usage into your server's workflow. With this in place, calls to `statsperso.free.fr`, `ftpperso.free.fr`, etc., will think you're connecting from metropolitan France – restoring full access to Free's hosting services **transparently** to your applications.

Sources:

- Free.fr personal pages FTP access policy (VPN requirement for non-French IPs) [1]
- Free.fr service domains (FTP, stats, etc.) and endpoints [2] [5]
- Fedora router configuration (DNS, DHCP, forwarding, NAT, WireGuard setup) [4] [3]
- WireGuard AllowedIPs usage for full-tunnel routing [6]
- Note on FTP and multiple connections (importance of routing by address) [8]

---

1 Les conditions d'accès au FTP des pages perso
http://les.pages.perso.chez.free.fr/les-conditions-d-acces-au-ftp-des-pages-perso.io

2 umowt.md
https://github.com/spectra-gallery/arquolab-sandbox-models/blob/76a0e49ec42b9aa5eeff308559ba59ce665a0b53/IDEO/umowt.md

3 4 7 apply-fedora.sh
https://github.com/spectra-gallery/arquolab-sandbox-models/blob/76a0e49ec42b9aa5eeff308559ba59ce665a0b53/uniphilabs/ark/exosysmu/scripts/apply-fedora.sh

5 212.27.63.0/24 IP Range - IPinfo.io
https://ipinfo.io/ips/212.27.63.0/24

6 10 11 My Whitehat Hacking Lab
https://www.mcnulty.blog/posts/whitehat-lab

8 Route FTP through WireGuard - Beginner Basics - MikroTik community forum
https://forum.mikrotik.com/t/route-ftp-through-wireguard/160217

9 wg0.conf
https://github.com/spectra-gallery/arquolab-sandbox-models/blob/76a0e49ec42b9aa5eeff308559ba59ce665a0b53/uniphilabs/ark/exosysmu/configs/wireguard/wg0.conf