Here is how I accomplished split tunneling with WireGuard and 3proxy

Intro:

Although the instructions below work for my setup, it's not a method that should be relied on if you have high security concerns. I needed to use split tunneling in Wireguard windows. I had one or two applications that I wanted to route through the Wireguard gateway, but everything else I wanted to route through my regular gateway.

By default, Wireguard will create a default route setting it as high priority (low metric). This forces all windows traffic out the WireGuard tunnel. You can delete the default route, but then you'd have to add a route each time the tunnel is connected, which is not convenient! Moreover, Wireguard by itself will not create a local socks5 so another program is needed to create the socks5 proxy. In this case I use 3proxy.

This seemed like a hassle, so after some digging, I managed to come up with solution that fits my requirements it involves the following:

- Disable automatic default route creation
- Configure allowed IPs
- Enable the use of scripts
- Add post-up/pre-down commands
- 3proxy tiny free proxy server (https://github.com/3proxy/3proxy)
- Netch (Optional - abandoned project) (https://github.com/NetchX/Netch)
- Proxifyre (Optional) (https://github.com/wiresock/proxifyre)

I won’t go into the contents of the wireguard config files explicitly, but it is useful to know the following before I go into my explanation of how to achieved split tunneling with WireGuard:

Disable automatic default route creation:

In the `[Interface]` section of your tunnel config, add

```
Table = off
```

This informs WireGuard not to create a default route automatically.

Configure allowed IPs:

You’ll probably want to set this to all IPs. `AllowedIPs` tells WireGuard what IP addresses to route through the tunnel. So if your application connects to various IPs, or you aren’t sure, perform this step. (After this we will fix the default routes so packets don’t actually go out through the tunnel.) If you know what you are doing, change the `AllowedIPs` to fit your requirements.

In the `[Peer]` section of your tunnel config, set this

```
AllowedIPs = 0.0.0.0/0
```

Enable the use of scripts:

Scripts are not enabled in WireGuard Windows by default. This needed to enable scripts for this method to work properly and set the correct route. To do this, you'll need to enable the `DangerousScriptExecution` registry key.

Open the registry editor (regedit.exe) and create a new `WireGuard` key in the following address:

```
HKEY_LOCAL_MACHINE\SOFTWARE\WireGuard
```

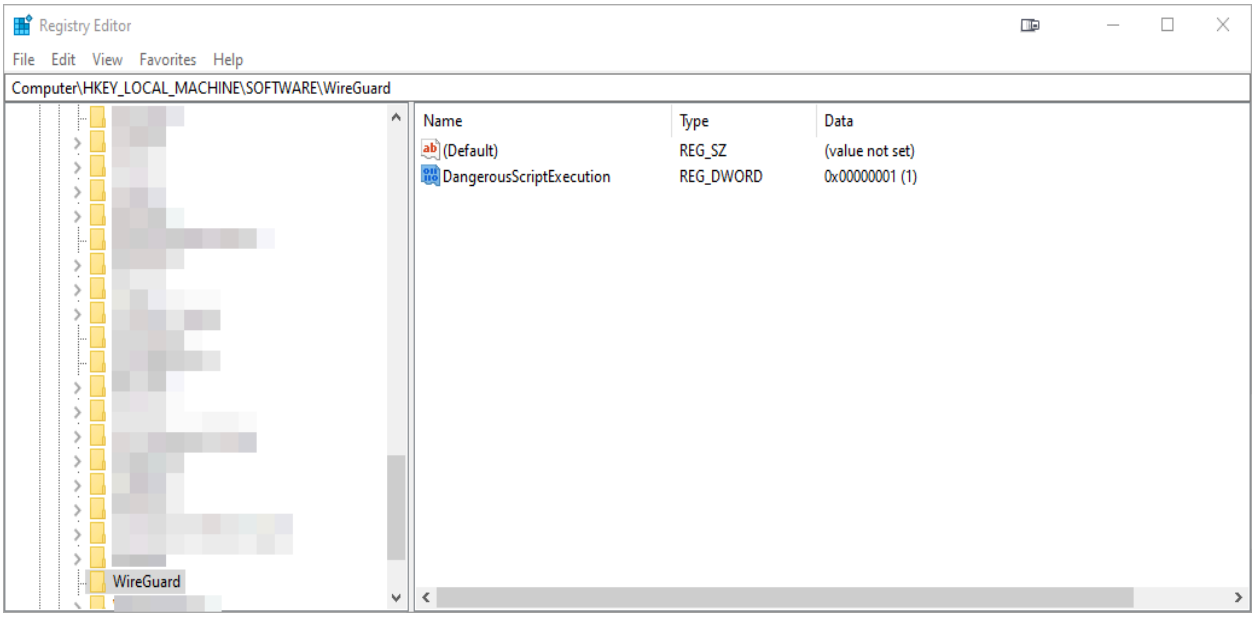
Create a new `DWORD` value in the new created key named:

```
DangerousScriptExecution
```

And set its value to:

```
1
```

It should look like this



Registry Editor

Add post-up/pre-down commands:

In the `[Interface]` section of your tunnel config, add these command lines pointing to the scripts location in windows. The last line tells WireGuard not to automatically create a default route when connected:

- `PostUp = c:\`**`(directory where is the file)`**`\up.cmd`
- `PreDown = c:\`**`(directory where is the file)`**`\down.cmd`
- `Table = off`

You will need to create the two files mentioned previously in the directory of your choice using notepad and make sure the extension is .cmd not .txt

(up.cmd) PS:make sure all the commands are in same line

```
powershell -command "$wgInterface = Get-NetAdapter -Name
%WIREGUARD_TUNNEL_NAME%; route add 0.0.0.0 mask 0.0.0.0
0.0.0.0 IF $wgInterface.ifIndex metric 9999;
Set-NetIPInterface -InterfaceIndex $wgInterface.ifIndex
-InterfaceMetric 9999;"

cd C:\3proxy\bin64
echo auth none> 3proxy-wireguard.conf
echo internal 127.0.0.1>> 3proxy-wireguard.conf
for /f "tokens=3 delims=: " %%I in ('netsh interface IPv4 show
addresses "%WIREGUARD_TUNNEL_NAME%" ^| findstr /C:"IP
Address"') do echo external %%I>> 3proxy-wireguard.conf
echo socks>> 3proxy-wireguard.conf
echo proxy>> 3proxy-wireguard.conf
start /b 3proxy.exe 3proxy-wireguard.conf
```

(down.cmd) PS:make sure all the commands are in same line

```
powershell -command "$wgInterface = Get-NetAdapter -Name
%WIREGUARD_TUNNEL_NAME%; route delete 0.0.0.0 mask 0.0.0.0
0.0.0.0 if $wgInterface.ifIndex metric 9999;
Set-NetIPInterface -InterfaceIndex $wgInterface.ifIndex
-InterfaceMetric 9999;"

taskkill /f /im 3proxy.exe
```

This will tell WireGuard to add a new default route after the tunnel is up (`PostUp`) with a metric of `9999` (low priority), as well as setting the metric of the interface

itself to `9999`. That means although Windows will technically route traffic through the tunnel, it will [probably] never happen because all other routes precede it in priority. As a comparison, most route metrics are less than 500 (lower is higher priority).

If you'd like to see your routes, open PowerShell and type

```
route print
```

Once the tunnel disconnects, it will delete that route (`PreDown`).

3proxy tiny free proxy server

On the application side, routing specific wireguard traffic using 3proxy, The IP assigned IP address by the WireGuard provider is then copied to a text file. This allows it to bind that IP address of the WireGuard interface with the 3proxy program to create a local Socks5 in windows.

In the documentation of 3proxy. The default settings will create two ports (This can be changed to create one depending on the options required) one is socks started on 127.0.0.1:1080 and the other is an HTTP proxy on 127.0.0.1:3128 so you have the choice to use whatever port you like locally.

You will need to download and extract 3proxy to a directory of your choice. (c:\3proxy) as an example.

I would recommend that you add the 3proxy directory in the exclusion list of windows antivirus as it often flags 3proxy as a virus and deletes it.

Netch (Optional)

I use this application to tunnel any program that does not support changing proxy

```
Step 1: Run the application Netch.exe downloaded from github.
Step 2: Click Server >> Add [Socks5] Server.
Step 3: Put any name you like in remarks, in Address:
127.0.0.1 and 1080 in the next box and save.
Step 4: In Mode, Select Discord. This is an example. You can
find or add any Application name .exe and just add it to
Netch.
Step 5: click Start
Step 6: Start Discord as you normally would
```

Final thoughts

If you don't trust the 3proxy and Netch (abandoned project, but still works! if you want a UI) method then you can use any other application you want to route through the tunnel that does not support binding to a specific interface or IP address, consider using ForceBindIP (free – not recommended) or **Proxifyre** (which is an opensource project on github) and a replacement of Netch.

Proxifier is ok, but not great! Not only can you specify specific applications to route through specific interfaces, but it also supports SOCKS5 and other proxies. You can even specify certain hostnames or ports to re-route traffic for local machine.