Scenario

We have "Desktop A" that is connected to a VPN and we want "Desktop B" to be able to access that VPN through "Desktop A"

- We have Windows "Desktop A" were our account has no admin rights
- We have Linux "Desktop B" were we have full control
- All incoming connections are blocked on "Desktop A"
- "Desktop A" does have git bash installed which comes with ssh.

Reverse SSH Proxy Tunnel

On "Desktop A":

```
ssh -fN -R localhost:1080 username@desktopb
```

What if you don't trust "Desktop A"

It might be unwise to do this method if you completly untrust "Desktop A" however there are steps you can take to limit it's access.

On "Desktop B"

```
useradd sshtunnel -m -d /home/sshtunnel -s /bin/false
```

On "Desktop B" edit `/etc/ssh/sshd_config`

```
Match User sshtunnel
   AllowTcpForwarding remote
   #GatewayPorts yes  # If you want more than just localhost to access proxy
   X11Forwarding no
   PermitTTY no
   ForceCommand /bin/false
```

You would probably want to add public key from "Desktop A" into `~/home/sshtunnel/.ssh/authorized_keys`
What about if there is a "Desktop C" that is like "Desktop B"?

Just do the steps Reverse SSH Proxy Tunnel and but going to "Desktop C" instead of "Desktop B". No need to change any port numbers either.

App Proxy Configuration

Most tools and applications can be configured individually to use a proxy. See below for ways section after this one if interested in ways to not have to individually set the proxy.

Browser

For browsers, you probably want to make a new profile with the proxy set to use "Desktop A". A handy tool for firefox is Firefox Multi-Account Containers it lets you have tabs of different profiles with different proxies set.

SSH

connect-proxy

Want to ssh through the VPN? Install connect-proxy

```
ssh -o ProxyCommand="connect-proxy -S localhost:1080 %h %p" user@destination
```

If there is a destination that you know you will always will need connect this way you can add it to ~/.ssh/ssh_config

```
Host somehost
  ProxyCommand connect-proxy -S localhost:1080 %h %p
```

netcat

There are a few different versions of netcat so see the manpage for your version

```
man nc
```

OpenBSD netcat

```
Host somehost
  ProxyCommand nc -X 5 -x localhost:1080 %h %p
```

nmap netcat
Curl or other software

You can set environment variable.

```sh
echo ALL_PROXY="socks5h://localhost:1080"
curl https://someurl_on_other_side
```

Node or http proxy

Node (or at least some older versions) can't use socks proxy. So you can setup a socks to http proxy.


- apt-install privoxy
- Edit `/etc/config` change `listen` and add `forward-socks5` directives

```sh
# listen on 9051
listen-address localhost:9051
max-client-connections 1000

forward-socks5    /    127.0.0.1:1080  .
```

Be sure to add to `.npmrc` in your project folder you want to use the proxy in.

```txt
http-proxy=http://localhost:9051
https-proxy=http://localhost:9051/
```

Docker

- [https://docs.docker.com/network/proxy/](https://docs.docker.com/network/proxy/)

These only set environment variables to be used by individual apps.
Use Proxy without configuring per app

Proxychains


Can set ~/.proxychains/proxychains.conf

```ini
[ProxyList]
socks5 127.0.0.1 1080
```

Then run your command with proxychains in front.

```
proxychains npm ci or proxychains mvn ... , etc.
```

redsocks

You can use iptables and redsocks to make routing to proxy possible and transparent for system apps. If you wish it should even be possible to setup this all up on a small device and then just change your gateway to the small device.

- [https://github.com/darkk/redsocks](https://github.com/darkk/redsocks)
- Possible script to help automate it all -
  [https://gist.github.com/vitex/1287517/10b4b6f80b8036bf155901e25715564abd9a92a3](https://gist.github.com/vitex/1287517/10b4b6f80b8036bf155901e25715564abd9a92a3)

Desktop A run script

run_desktopb.bat:

```
"C:\Program Files\Git\bin\sh.exe" --login -i -c "~/run_desktopb.sh"
```

run_desktopb.sh

```
#!/bin/bash
wmic process get commandline /format:list | grep "[:1080 username@desktopb]" ||
```
VPN Settings on Desktop A

Anyconnect

If Anyconnect is the VPN on "Desktop A", make sure you enabled **Allow local LAN access when using VPN (if configured).**

Proxy device

If you end up using a proxy device, make sure you do the following:

- Make sure `GatewayPorts yes` is enabled in `/etc/sshd_config` for sshtunnel
  - `ssh -R command change localhost to * like: ssh -fN -R *:1080 user@desktopb`
- If using privoxy change `/etc/privoxy/config` to not just listen on localhost, so: `listen :9051`
- Be sure to use the IP address for the device in your app proxy settings.

Security

Just allow the IP address that should have access.

```bash
export IT=iptables
export EIF=eth0

$IT -t nat -A POSTROUTING -o $EIF -j MASQUERADE
$IT -t nat -A PREROUTING -p tcp -i $EIF --dport 1080 -j DNAT --to-destination 127.0.0.1:1080

for lan_ip in 192.168.x.B 192.168.x.C; do
  $IT -A INPUT -p tcp -i $EIF --dport 1080 -s ${lan_ip} -j ACCEPT
  $IT -A INPUT -p tcp -i $EIF --dport 9051 -s ${lan_ip} -j ACCEPT
done

$IT -A INPUT -p tcp -i $EIF --dport 1080 -j DROP
$IT -A INPUT -p tcp -i $EIF --dport 9051 -j DROP
```
Table of Contents

- Scenario
- Reverse SSH Proxy Tunnel
  - What if you don't trust "Desktop A"
  - What about if there is a "Desktop C" that is like "Desktop B"?
- App Proxy Configuration
  - Browser
  - SSH
    - connect-proxy
    - netcat
  - Curl or other software
  - Node or http proxy
  - Docker
- Use Proxy without configuring per app
- Proxychains
- redsocks
- Desktop A run script
- VPN Settings on Desktop A
  - Anyconnect
- Proxy device
  - Security