How to evade VPN disruption

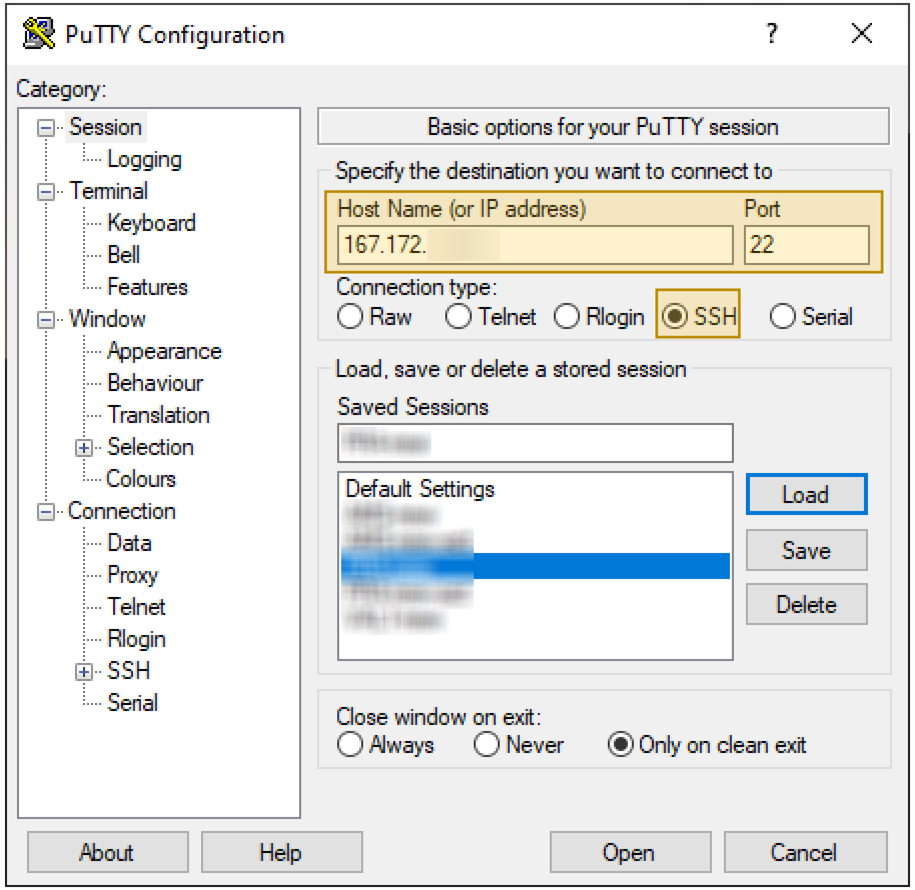
with SSH proxy tunnels

Create a minimal Ubuntu / FreeBSD VM on Digital Ocean (or similar, i.e. Vultr)

Create and associate a certificate to the VM's SSH server (better with password) for the tunnel user (not root)

Patch, harden and firewall the VM. You can also set the SSH server to listen on a high port and not on port 22 and close all other ports / stop all other services for better performances and security.

To begin, create a socks5 proxy tunnel via SSH from the client to the remote VM.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 167.172.[redacted]

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

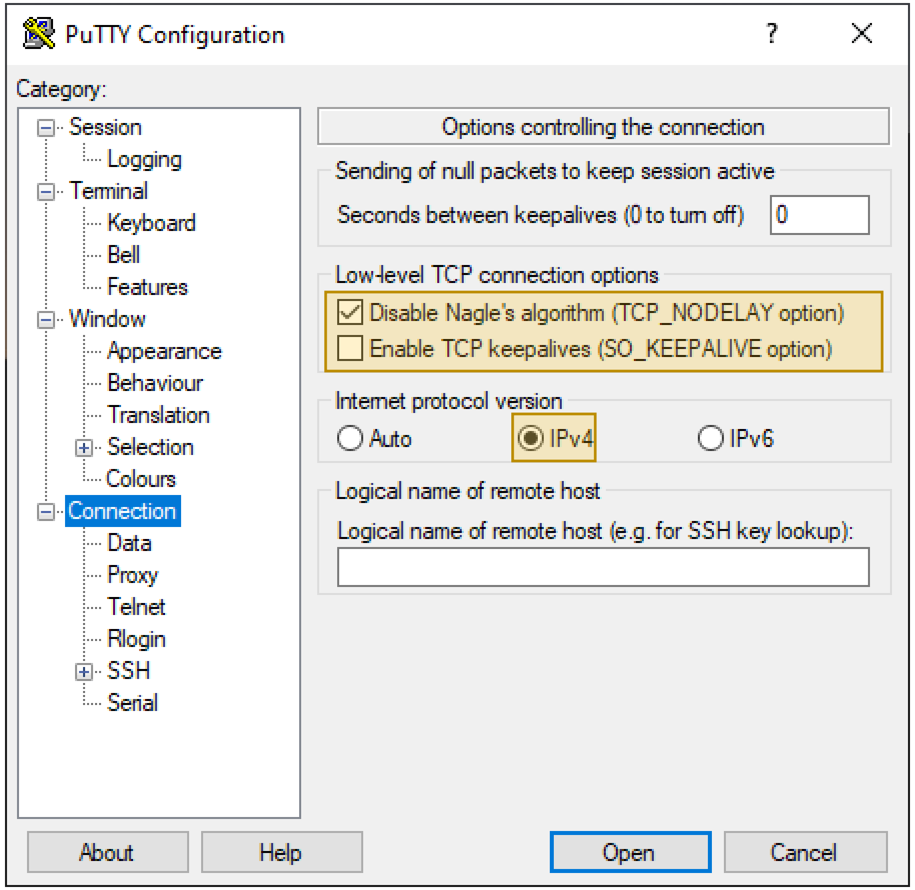
Saved Sessions: [redacted]

Default Settings
[redacted]

Close window on exit: ○ Always ○ Never ● Only on clean exit

To improve the performances of the proxy tunnel:

- disable the Nagle’s algorithm
- do not enable TCP keepalives
- Use IPv4 only



PuTTY Configuration

Options controlling the connection

Sending of null packets to keep session active

Seconds between keepalives (0 to turn off) 0

Low-level TCP connection options

☑ Disable Nagle's algorithm (TCP_NODELAY option)

□ Enable TCP keepalives (SO_KEEPALIVE option)

Internet protocol version

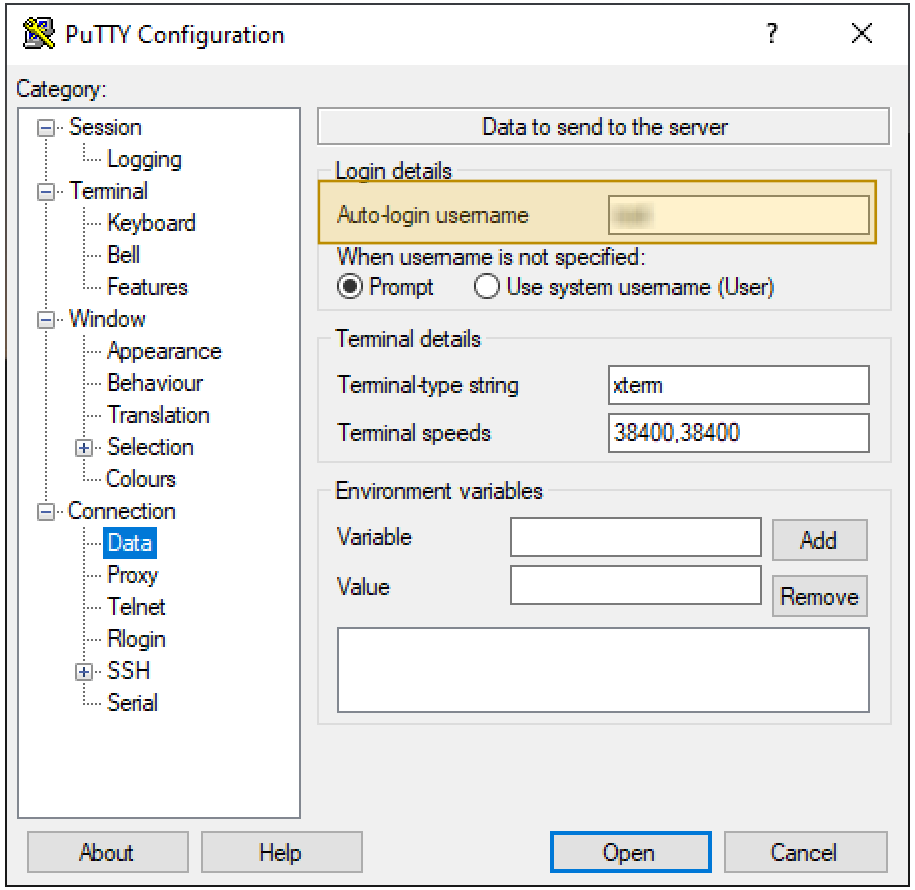
○ Auto ● IPv4 ○ IPv6

Logical name of remote host

Logical name of remote host (e.g. for SSH key lookup):

To improve the security of the proxy tunnel:

- Create a dedicated, non admin user on the VM
- Disable SSH access for root
- In auto login username, input the tunnel user



Data to send to the server

Login details

Auto-login username [redacted]

When username is not specified:

● Prompt ○ Use system username (User)

Terminal details

Terminal-type string: xterm

Terminal speeds: 38400,38400

Environment variables

Variable

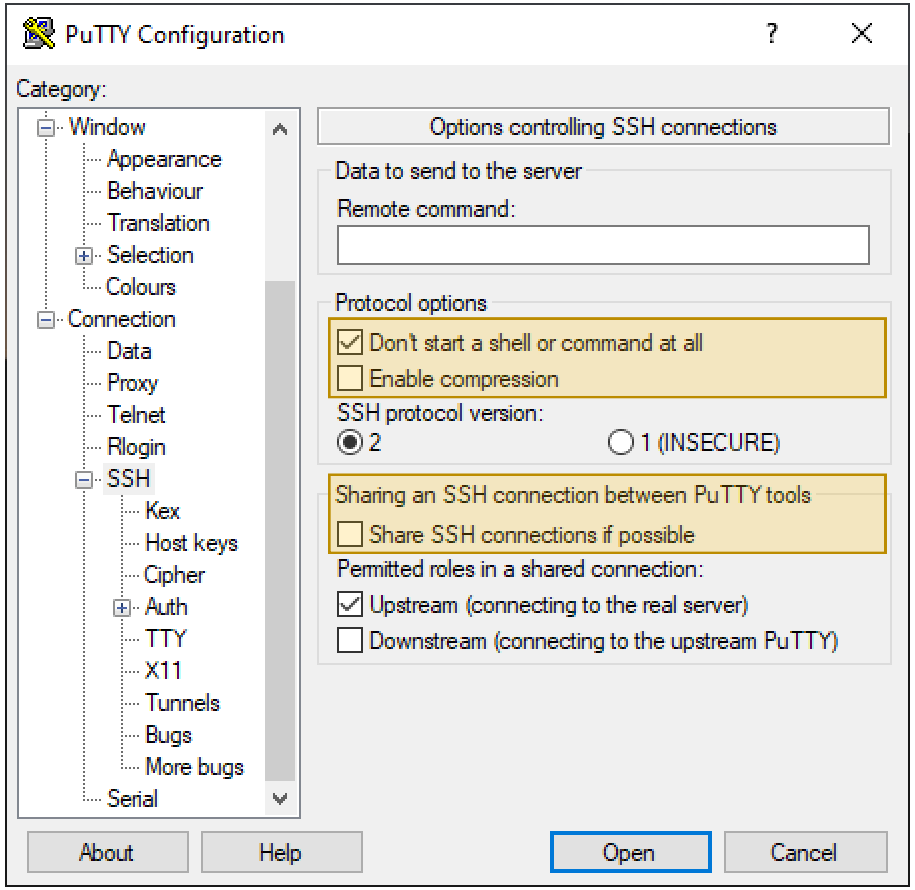
Value

To improve the security of the proxy tunnel:

- Don’t start a shell
- Do not permit shared connections among Putty tools (on Windows, if using Putty as SSH client)

To improve the performances:

- Do not enable the SSH compression (you will be tunneling encrypted traffic all the time)



PuTTY Configuration

Category:

Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

☑ Don't start a shell or command at all

□ Enable compression

SSH protocol version:

● 2 ○ 1 (INSECURE)

Sharing an SSH connection between PuTTY tools

□ Share SSH connections if possible

Permitted roles in a shared connection:

☑ Upstream (connecting to the real server)

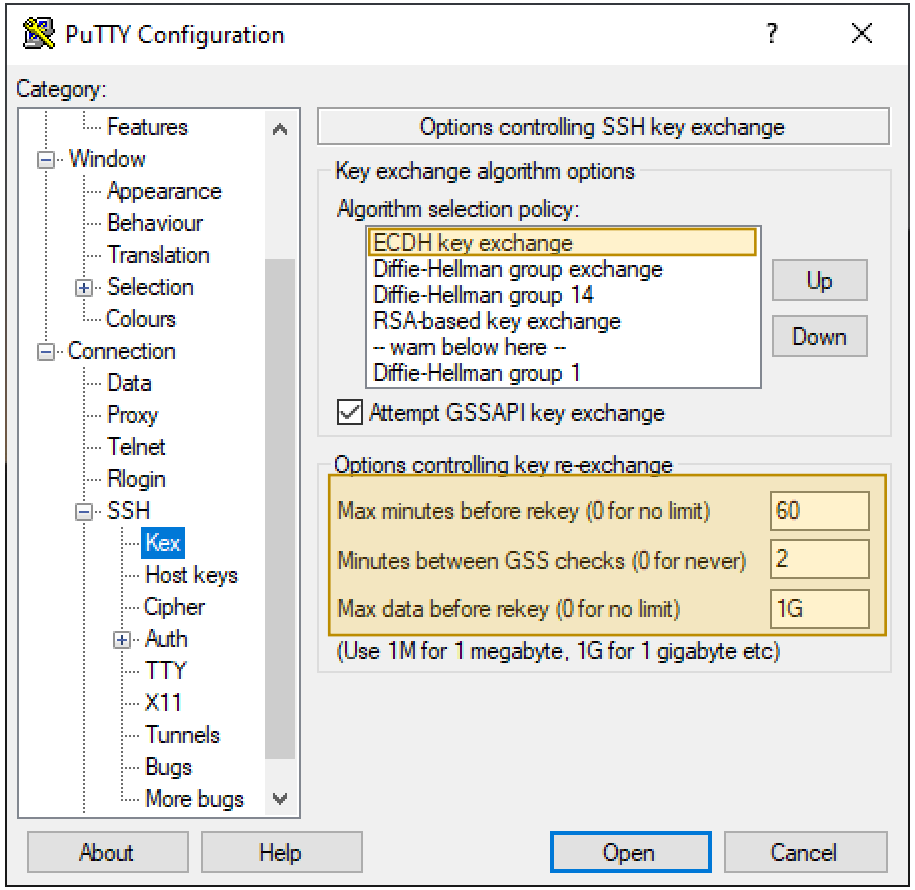
□ Downstream (connecting to the upstream PuTTY)

To improve the security of the proxy tunnel:

- Use ECDH key exchange

- Re-key frequently. Depending on how much traffic you make, adjust the “max minutes” / “max data” values accordingly.

  I.e. if you use a lot of video streaming / vdc or share big amounts of data, you can lower the “max minutes” to 45, etc.



Options controlling SSH key exchange

Key exchange algorithm options

Algorithm selection policy:

ECDH key exchange
Diffie-Hellman group exchange
Diffie-Hellman group 14
RSA-based key exchange
-- warn below here --
Diffie-Hellman group 1

☑ Attempt GSSAPI key exchange

Options controlling key re-exchange

Max minutes before rekey (0 for no limit) 60

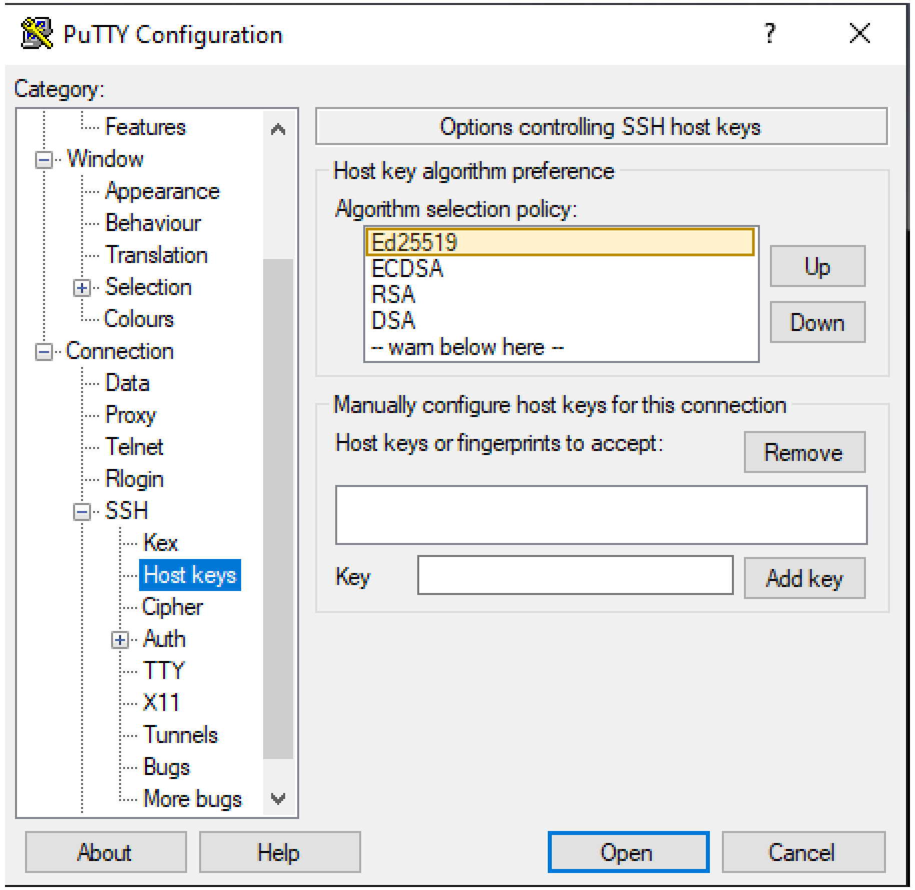
Minutes between GSS checks (0 for never) 2

Max data before rekey (0 for no limit) 1G

(Use 1M for 1 megabyte, 1G for 1 gigabyte etc)

To improve the security of the proxy tunnel:

- Ed25519 is *better*.
- Make sure to generate an Ed25519 certificate for the SSH server
- ECDSA is also “ok”, but don’t use RSA!



PuTTY Configuration

Options controlling SSH host keys

Host key algorithm preference

Algorithm selection policy:

Ed25519
ECDSA
RSA
DSA
-- warn below here --

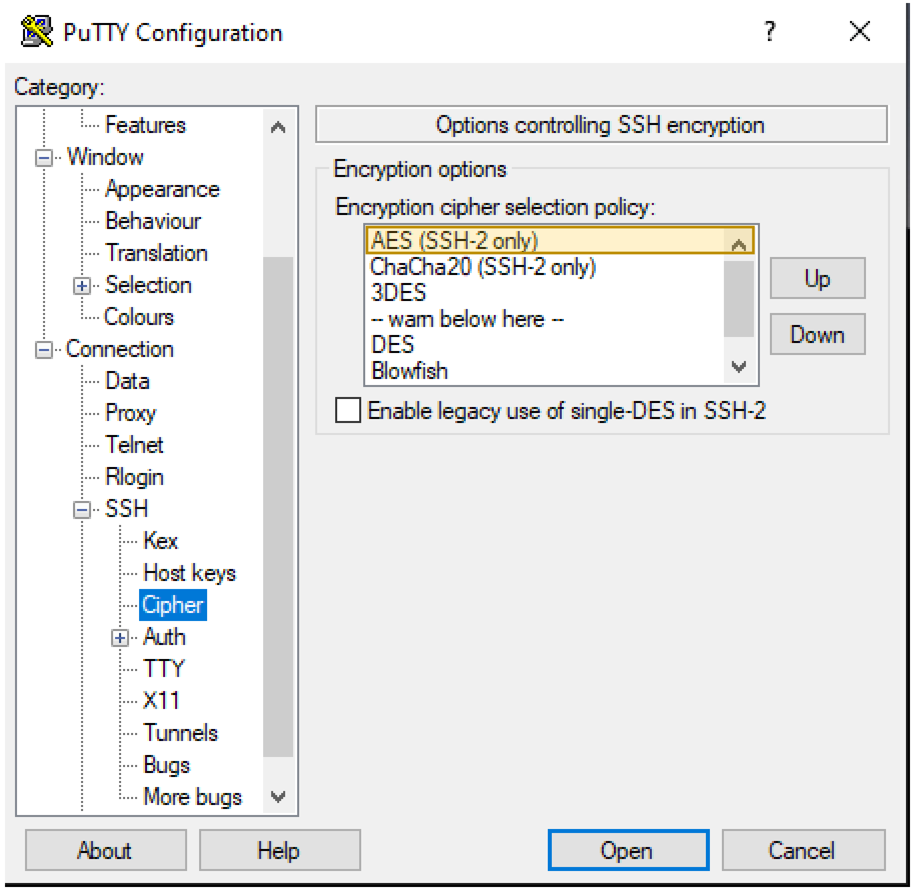
Manually configure host keys for this connection

Host keys or fingerprints to accept:

Key

To improve the security and the performance of the SSH proxy tunnel:

- Use AES. It is accelerated in hardware on all modern CPUs and operating systems.
- ChaCha20 is also “ok” from a security point of view, but it is not accelerated (= more latency).
- Don’t use 3DES!



PuTTY Configuration

Options controlling SSH encryption

Encryption options

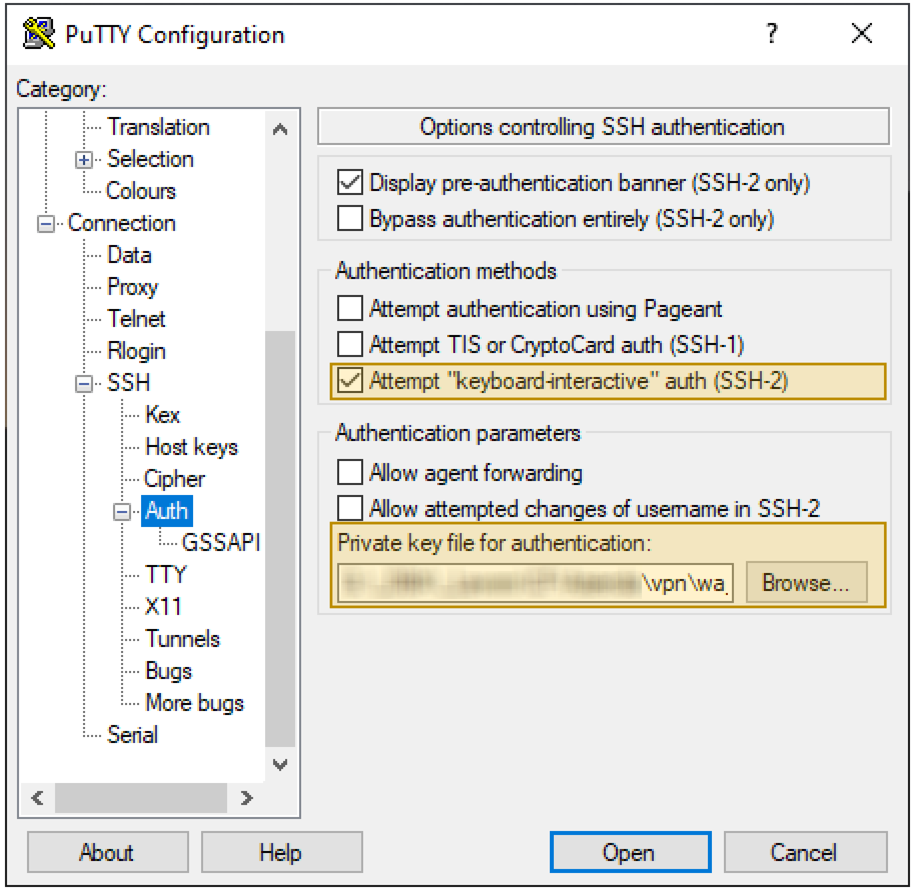
Encryption cipher selection policy:

AES (SSH-2 only)
ChaCha20 (SSH-2 only)
3DES
-- warn below here --
DES
Blowfish

□ Enable legacy use of single-DES in SSH-2

To improve the security of the proxy tunnel:

- Only allow "keyboard-interactive" auth.
  This is required if the certificate you created for the VM has a password (highly recommended).
- Keep the private key file in a safe location on the computer (i.e. an encrypted partition)



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

☐ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

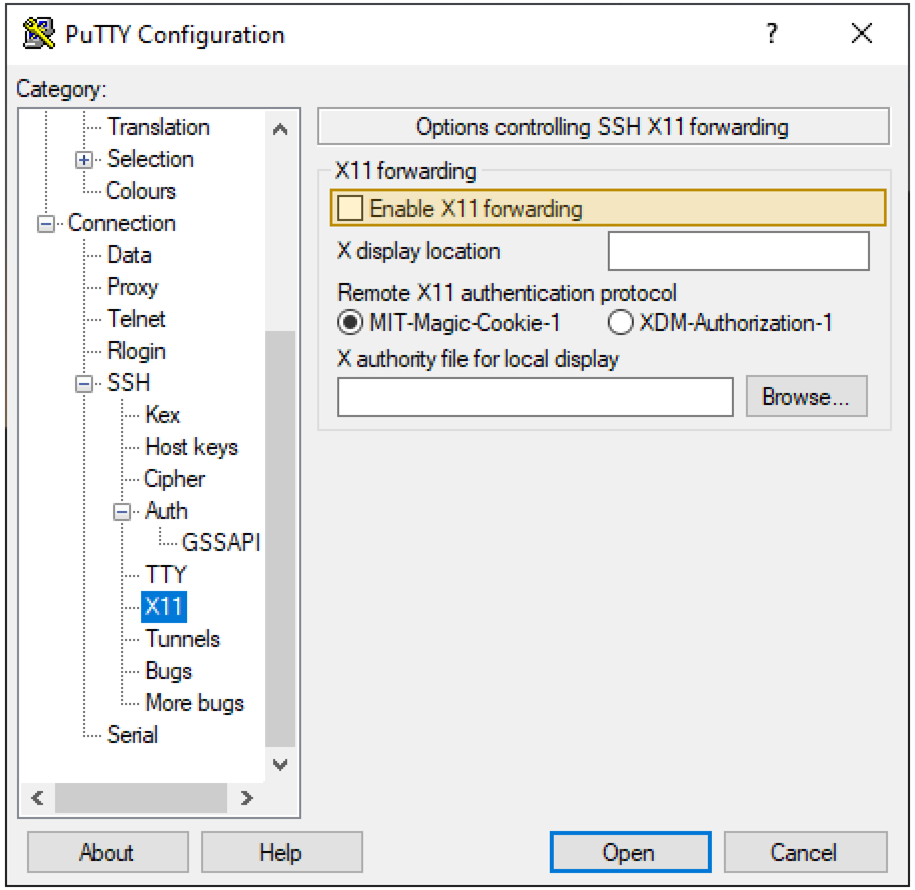
Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:
[redacted]\vpn\wa Browse...

To improve the security of the proxy tunnel:

- Do not activate X11 forwarding



PuTTY Configuration

Category:

Options controlling SSH X11 forwarding

X11 forwarding

□ Enable X11 forwarding

X display location

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

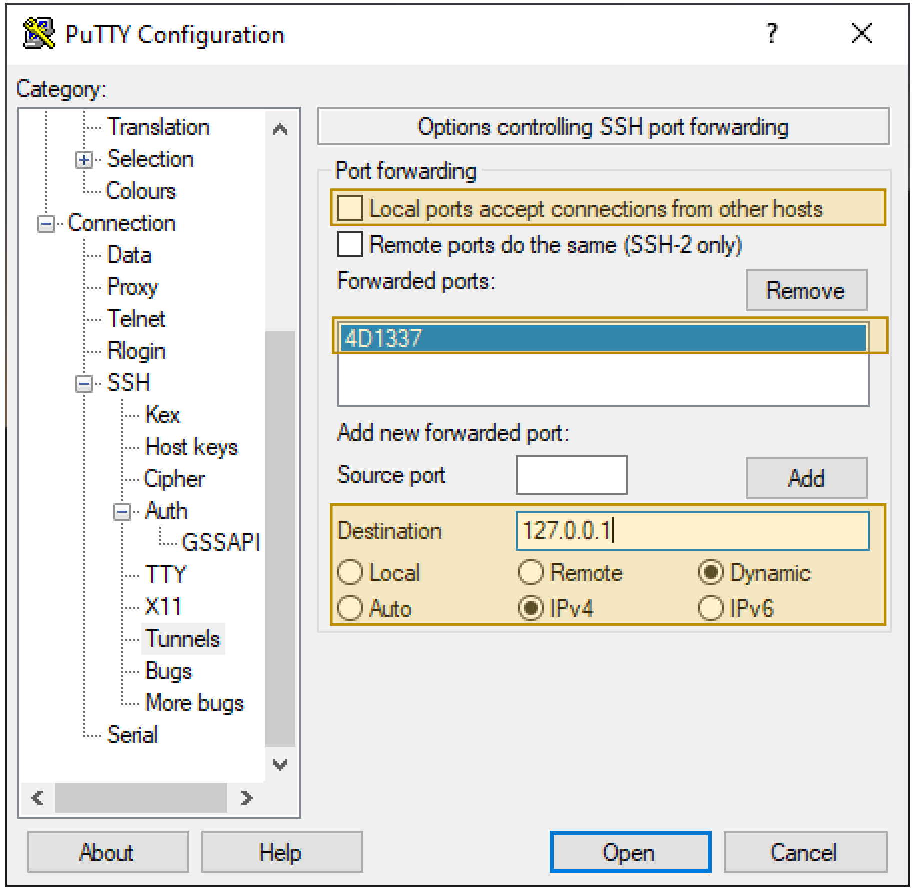
X authority file for local display

To improve the security of the proxy tunnel:

- Do not accept connections from other hosts (only from localhost)

Create a Dynamic IPv4 socks5 proxy tunnel, listening on a port of your choice (in this case, 1337) on 127.0.0.1

Remember to save the SSH client configuration before running it ☺



Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

4D1337

Add new forwarded port:

Source port

Destination 127.0.0.1

○ Local ○ Remote ● Dynamic

○ Auto ● IPv4 ○ IPv6

For the OpenVPN connection that you want to tunnel, remember to use TCP and not UDP, as it cannot be tunneled. If you use an UDP based VPN, *it will not work*.

Do not activate the VPN compression.

To improve the security of the proxy tunnel:

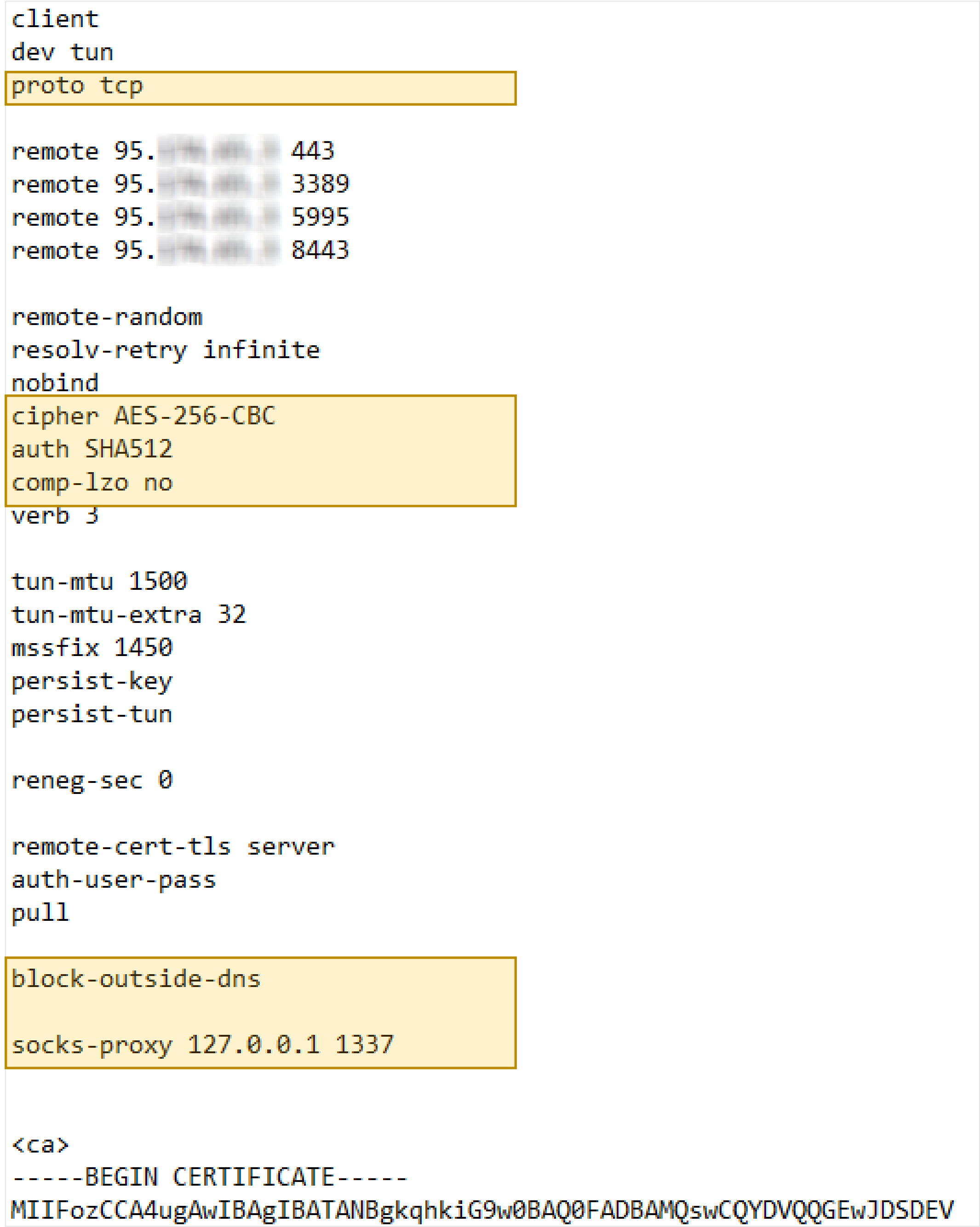
- Use AES-256-CBC
- Use SHA512
- Block outside DNS resolution to avoid DNS leaks (important!)

Define the SSH socks5 proxy, listening on 127.0.0.1 [port].

Done! Start the SSH tunnel and when it's up, start the VPN. The VPN will go into the tunnel and become invisible to the ISP.

For added security, you can rebuild your VM exit node(s) frequently.

You can even use more than one tunnel at a time, for different VPNs, by using different listening ports on localhost.



```
client
dev tun
proto tcp

remote 95.[redacted] 443
remote 95.[redacted] 3389
remote 95.[redacted] 5995
remote 95.[redacted] 8443

remote-random
resolv-retry infinite
nobind
cipher AES-256-CBC
auth SHA512
comp-lzo no
verb 3

tun-mtu 1500
tun-mtu-extra 32
mssfix 1450
persist-key
persist-tun

reneg-sec 0

remote-cert-tls server
auth-user-pass
pull

block-outside-dns

socks-proxy 127.0.0.1 1337


<ca>
-----BEGIN CERTIFICATE-----
MIIFozCCA4ugAwIBAgIBATANBgkqhkiG9w0BAQ0FADBAMQswCQYDVQQGEwJDSDEV
```