**Cloaking, Obfuscation & Remote Login**

**These techniques are used to encrypt traffic to avoid content filters and alter traffic so it looks like allowable traffic**

**Creates a TLS/SSL encryption wrapper between remote client and remote server**

Having no SSL aware daemons on the system, they can easily be set up to communicate with clients over encrypted channels

* Wrapping VPN traffic in an S tunnel (SSL tunnel) makes it a genuine VPN/encrypted connection as using OpenVPN or something like Nord is obvious

Apt-get install stunnel4

You can tunnel SSH connection through SSL using Stunnel too on Debian/Ubuntu

**Obfsproxy**

Can be used for more than Tor, can be used for any traffic

Pluggable transport

Disguises traffic

**Psiphon**

**Lahana**

VPN 🡪 Tor

DNScat2 / Iodine DNS Tunnel

* Tunnel through DNS
* VPN over DNS

**Remote Login**

**Using a remote login tool, SSL encrypt it and SSH it through to your home server via an allowed port like 443 or 80 (local port forwarding)**

SSH -L 8080:localhost:5900 -p 443 [root@demo.stationx.net](mailto:root@demo.stationx.net)

Create local SSH port 8080, connect to port 5900 which is where the remote login service is running

-p 443, tunnels through to 443 to bypass firewall

This will SSH out through 443 to connect to the server which will then redirect to port 5900 on the server where you will be able to access the remote login tool

**Remote Desktop in a browser**

* Apache Guacamole