========== SERVIDOR (VM de Azure) ==========

NOTA: Abrir en la VM los puertos 51820 (UDP) y 8888 (TCP). El primero para Allow-WireGuard y el segundo para Allow-Obfs (ofuscacion).

Una vez creada la maquina virutal, y cuando Azure nos ha dado la clave privada, al conectarnos por primera vez para conectarse por ssh ejecutar:

ssh -i vpn_key.pem azureuser@68.221.160.90

es posible que de un error de permisos. Para solucionarlo:

chmod 600 vpn_key.pem

Crear el archivo vpn.sh:

#!/bin/bash

Actualizar sistema

sudo apt update && sudo apt upgrade -y

Instalar WireGuard y simple-obfs

sudo apt install wireguard simple-obfs -y

Generar claves

wg genkey | tee server_private.key | wg pubkey > server_public.key wg genkey | tee client_private.key | wg pubkey > client_public.key

Crear directorio de configuración

sudo mkdir -p /etc/wireguard sudo mv server_private.key server_public.key client_private.key client_public.key /etc/wireguard/

Configurar wg0.conf

sudo tee /etc/wireguard/wg0.conf > /dev/null <<EOF [Interface] Address = 10.0.0.1/24 ListenPort = 51820 PrivateKey = $(cat /etc/wireguard/server_private.key) PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer] PublicKey = $(cat /etc/wireguard/client_public.key) AllowedIPs = 10.0.0.2/32 EOF

Activar IP forwarding

sudo sysctl -w net.ipv4.ip_forward=1 echo "net.ipv4.ip_forward=1" | sudo tee -a /etc/sysctl.conf

Iniciar WireGuard

sudo systemctl enable wg-quick@wg0 sudo systemctl start wg-quick@wg0

Ejecutar proxy simple-obfs en segundo plano

nohup simple-obfs -s -l 8888 --obfs http > /dev/null 2>&1 &

echo " ✅ WireGuard + proxy ofuscado instalados y ejecutándose en el puerto 8888."

Luego ejecutarlo:

chmod +x vpn.sh

sudo ./vpn.sh

Comprobar y verificar que la interfaz de red en el script es eth0. Ejecutar:

ip a

=======Paso intermedio: Una vez ejecutado y comprobado todo lo anterior:========

Obtén la clave privada del cliente y la clave pública del servidor

Estas están en la VM, en /etc/wireguard/. Ejecutar:

sudo cat /etc/wireguard/client_private.key
sudo cat /etc/wireguard/server_public.key

Guarda el contenido de cada una en tu máquina local, lo necesitarás para configurar el cliente.

==========Configurar el Cliente: ==========

Configura el cliente WireGuard

Crea un archivo de configuración en tu dispositivo cliente (PC, móvil, etc.), por ejemplo wg0.conf con este contenido (reemplaza las claves y la IP pública de tu VM):

```
[Interface]
PrivateKey = IPXCanj4HeBzSPyv2o+1HUJ6KU++yeFXk/bIncNvS2o=
Address = 10.0.0.2/24
DNS = 1.1.1.1


[Peer]
PublicKey = PyGR+wud1RWLr/qBOehhBsSMJD6By0ihGWlWdoTmo3A=
Endpoint = 68.221.160.90:51820
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 25
```

QR: VPN sin ofuscación

==========OFUSCACIÓN==========

===Script de instalación de shadowsocks (en el servidor, osea la maquina virtual)===

```
#!/bin/bash

# ACTUALIZAR SISTEMA
sudo apt update && sudo apt upgrade -y

# INSTALAR Shadowsocks-libev y simple-obfs
sudo apt install shadowsocks-libev simple-obfs -y

# CREAR CONFIGURACIÓN
sudo tee /etc/shadowsocks-libev/config.json > /dev/null <<EOF
{
```

"server":"0.0.0.0",

"server_port":8888,

"local_port":1080,

"password":"tuseguro123",

"timeout":300,

"method":"aes-256-gcm",

"plugin":"obfs-server",

"plugin_opts":"obfs=http"

}

EOF

 # HABILITAR E INICIAR EL SERVICIO

sudo systemctl enable shadowsocks-libev

sudo systemctl restart shadowsocks-libev

echo "✅ Shadowsocks + simple-obfs están funcionando en el puerto 8888"

NOTA: Se puede verificar que el puerto 8888 está escuchando con el comando:

sudo lsof -i -P -n | grep LISTEN

Para monitorizar

Y luego en el cliente (en el movil):

◇ Si usas Android

1. **Instala:**
    a. App: Shadowsocks Android (o desde Play Store si disponible)
    b. Plugin: simple-obfs plugin APK
2. En la app Shadowsocks:

a. Tipo: Shadowsocks
b. Servidor: <TU_IP_PUBLICA_DE_LA_VM>
c. Puerto: 8888
d. Método: aes-256-gcm
e. Contraseña: tuseguro123
f. Plugin: **habilita el plugin**
g. Plugin configuration:

obfs=http

3. Guarda y conecta.

En la propia aplicación puedes hacer una prueba de conexión.

Es importante conectarse usando la aplicación de shadowsocks **(vpn con ofuscación).** NO CONECTARSE NUNCA DESDE CHINA A LA VPN SIN OFUSCACIÓN.

◇ Si usas iPhone (iOS)

1. **Instala la app Shadowrocket** (pago único, pero muy confiable).
2. Abre la app y añade un nuevo servidor manualmente:

Yaml
CopyEdit

Type: Shadowsocks
Server: <TU_IP_PUBLICA>
Port: 8888
Encryption: aes-256-gcm
Password: tuseguro123
Obfuscation: HTTP
Obfuscation Host: www.bing.com

- Opcional: marca “UDP relay” si lo deseas.
- Guarda y conecta. Si todo está bien, verás tráfico saliendo por tu túnel seguro.

NOTA: Para monitorear conexiones se puede ejecutar este comando en la VM:

sudo ss -tunap | grep 8888

NOTA: Útil para ver las peticiones que se realizan al servidor (VM):

sudo tcpdump -n -i any port 8888

NOTA: Con el siguiente comando:

sudo tcpdump -n -A -i eth0 port 8888

Se pueden ver peticiones de la forma:

.. ..aiwGET / HTTP/1.1

Host: cloudfront.net:8888

User-Agent: curl/7.39.0

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: PV5HUIsuSPLkVaOZ2QZ6EQ==

De esta forma, nos hemos asegurado con evidencia visual de que la ofuscación funciona correctamente.

==========v2ray + TLS (certbot)==========

Para mejorar la privacidad y el camuflaje, usamos v2ray con TLS. Primero, guardamos en un script v2ray.sh el siguiente código:

#!/bin/bash

CONFIGURACIÓN

DOMAIN="chinesetexts.duckdns.org" UUID=$(cat /proc/sys/kernel/random/uuid) V2RAY_CONFIG_PATH="/usr/local/etc/v2ray" WS_PATH="/myapp"

ACTUALIZA E INSTALA DEPENDENCIAS

apt update && apt upgrade -y apt install -y curl socat cron certbot unzip jq

INSTALA V2RAY

bash <(curl -Ls https://github.com/v2fly/fhs-install-v2ray/raw/master/install-release.sh)

OBTIENE CERTIFICADO TLS (modo standalone)

systemctl stop nginx apache2 2>/dev/null certbot certonly --standalone -d "$DOMAIN" --non-interactive --agree-tos -m tuemail@example.com

CREA CONFIG DE V2RAY

mkdir -p "$V2RAY_CONFIG_PATH"

cat > "$V2RAY_CONFIG_PATH/config.json" <<EOF { "log": { "access": "/var/log/v2ray/access.log", "error": "/var/log/v2ray/error.log", "loglevel": "warning" }, "inbounds": [{ "port": 443, "protocol": "vless", "settings": { "clients": [{ "id": "$UUID", "flow": "xtls-rprx-vision" }], "decryption": "none" }, "streamSettings": { "network": "ws", "security": "tls", "tlsSettings": { "certificates": [{ "certificateFile": "/etc/letsencrypt/live/$DOMAIN/fullchain.pem", "keyFile": "/etc/letsencrypt/live/$DOMAIN/privkey.pem" }] }, "wsSettings": { "path": "$WS_PATH" } } }], "outbounds": [{ "protocol": "freedom", "settings": {} }] } EOF

PERMISOS

mkdir -p /var/log/v2ray chown -R nobody:nogroup "$V2RAY_CONFIG_PATH"

HABILITA Y ARRANCA V2RAY

systemctl enable v2ray systemctl restart v2ray

MOSTRAR CONFIG CLIENTE

echo echo " ✅ V2Ray instalado y configurado correctamente" echo " 🔑 UUID: $UUID" echo " 🌐 Dominio: $DOMAIN" echo " 📍 WebSocket Path: $WS_PATH" echo " 📦 Puerto: 443" echo echo " ⚙️ Configuración para cliente:" echo cat <<EOL { "v": "2", "ps": "ChinaBypass", "add": "$DOMAIN", "port": "443", "id": "$UUID", "aid": "0", "net": "ws", "type": "none", "host": "$DOMAIN", "path": "$WS_PATH", "tls": "tls" } EOL

Importante poner un correo para certbot. Puede que de algunos errores al ejecutar el script. Tener en cuenta de que hay que abrir los puertos 80 y 443 de la VM.

Ahora establecemos la renovación automática de TLS:

echo "0 3 * * * root certbot renew --quiet && systemctl restart v2ray" | sudo tee -a /etc/crontab

Para probar una renovación automática manual:

sudo certbot renew --dry-run

==========Problema gordo con v2ray y tls (certbot)==========

Ver fichero: traza_v2ray y analizar la traza. Sobre todo tuvo que ver con problemas en el certificado y la clave generado por certbot, los archivos de configuración y las rutas configuradas.

QR con la configuración del cliente de v2rayNG

Agregar un script para reiniciar mi servicio v2ray una vez certbot renueva los certificados:

sudo nano /etc/letsencrypt/renewal-hooks/post/restart-v2ray.sh

Escribimos el script restart-v2ray.sh:

#!/bin/bash

systemctl restart v2ray

Y lo hacemos ejecutable:

sudo chmod +x /etc/letsencrypt/renewal-hooks/post/restart-v2ray.sh