紅襪隊在行動

當然可以！我將指導您如何使用您的 Mac 電腦設置 Shadowsocks 代理，並配置您的 OpenWRT 路由器，以將所有連接設備的流量通過此代理路由。此設置涉及以下主要步驟：

1. 在 macOS 上設置 Shadowsocks 客戶端（使用 Shadowsocks-NG 或 Clash）
2. 配置 macOS 以允許外部代理連接
3. 為您的 Mac 分配靜態 IP
4. 在 OpenWRT 上安裝和配置 Redsocks
5. 將 OpenWRT 的流量重定向通過 macOS 代理
6. 測試代理設置

讓我們詳細介紹每個步驟。

---

1. 在 macOS 上設置 Shadowsocks 客戶端

您可以使用 Shadowsocks-NG 或 Clash 作為您的 Shadowsocks 客戶端。以下是兩者的說明。

選項 A：使用 Shadowsocks-NG

Shadowsocks-NG 是 macOS 上流行且用戶友好的 Shadowsocks 客戶端。

步驟 1：下載並安裝 Shadowsocks-NG

1. 下載 Shadowsocks-NG：
   - 訪問Shadowsocks-NG GitHub Releases 頁面。
   - 下載最新的`.dmg`文件。
2. 安裝應用程序：
   - 打開下載的`.dmg`文件。
   - 將 ShadowsocksX-NG 應用程序拖到您的應用程序文件夾中。
3. 啟動 Shadowsocks-NG：
   - 從您的應用程序文件夾中打開 ShadowsocksX-NG。
   - 您可能需要在系統偏好設置中授予應用程序必要的權限。

步驟 2：配置 Shadowsocks-NG

1. 打開偏好設置：
    - 點擊菜單欄中的 ShadowsocksX-NG 圖標。
    - 選擇 “打開 ShadowsocksX-NG” > “偏好設置” 。
2. 添加新服務器：
    - 導航到 “服務器” 選項卡。
    - 點擊 “+” 按鈕以添加新服務器。
3. 導入 Shadowsocks URL：
    - 複製您的 Shadowsocks URL：

      ```
      ss://[ENCRYPTED_PASSWORD]@xxx.xxx.xxx.xxx:xxxxx/?outline=1
      ```

    - 導入方法：
        - 點擊 “導入” 。
        - 粘貼您的 Shadowsocks URL。
        - Shadowsocks-NG 應自動解析並填寫服務器詳細信息。
4. 設置本地代理：
    - 確保 “啟用 SOCKS5 代理” 已勾選。
    - 記下本地端口（默認通常為 `1080`）。
5. 保存並激活：
    - 點擊 “確定” 以保存服務器。
    - 切換 “啟用 Shadowsocks” 開關為 ON。

選項 B：使用 Clash

Clash 是一個支持多種協議（包括 Shadowsocks）的多功能代理客戶端。

步驟 1：下載並安裝 Clash

1. 下載 Clash for macOS：
    - 訪問Clash GitHub Releases 頁面。
    - 下載最新的 Clash for macOS 二進制文件。
2. 安裝應用程序：

- 將下載的 Clash 應用程序移動到您的應用程序文件夾中。

3. 啟動 Clash：
   - 從您的應用程序文件夾中打開 Clash。
   - 您可能需要在系統偏好設置中授予必要的權限。

步驟 2：配置 Clash

1. 訪問配置文件：
   - Clash 使用 YAML 配置文件。您可以使用 TextEdit 或 Visual Studio Code 等文本編輯器創建或編輯它。
2. 添加您的 Shadowsocks 服務器：
   - 創建一個配置文件（例如，`config.yaml`），內容如下：

```yaml
port: 7890
socks-port: 7891
allow-lan: true
mode: Rule
log-level: info

proxies:
  - name: "MyShadowsocks"
    type: ss
    server: xxx.xxx.xxx.xxx
    port: xxxxx
    cipher: chacha20-ietf-poly1305
    password: "xxxxxx"

proxy-groups:
  - name: "Default"
    type: select
    proxies:
      - "MyShadowsocks"
      - "DIRECT"

rules:
  - MATCH,Default
```

注意：

- `port` 和 `socks-port` 定義 Clash 將監聽的 HTTP 和 SOCKS5 代理端口。
- `allow-lan: true` 允許 LAN 設備使用代理。
- `proxies` 部分包括您的 Shadowsocks 服務器詳細信息。
- `proxy-groups` 和 `rules` 確定流量如何路由。

3. 使用配置啟動 Clash：
   - 啟動 Clash 並確保它使用您的 `config.yaml` 文件。
   - 啟動 Clash 時可能需要指定配置路徑。
4. 驗證代理是否運行：
   - 確保 Clash 已激活並連接到您的 Shadowsocks 服務器。
   - 檢查菜單欄圖標以獲取狀態。

---

2. 配置 macOS 以允許外部代理連接

默認情況下，Shadowsocks 客戶端將代理綁定到 `localhost`（`127.0.0.1`），這意味著只有 Mac 可以使用代理。要允許您的 OpenWRT 路由器使用此代理，您需要將代理綁定到 Mac 的 LAN IP。

對於 Shadowsocks-NG：

1. 打開偏好設置：
   - 點擊菜單欄中的 ShadowsocksX-NG 圖標。
   - 選擇 “打開 ShadowsocksX-NG” > “偏好設置” 。
2. 轉到高級選項卡：
   - 導航到 “高級” 選項卡。
3. 設置監聽地址：
   - 將 “監聽地址” 從 `127.0.0.1` 更改為 `0.0.0.0` 以允許來自任何接口的連接。
   - 或者，指定 Mac 的 LAN IP（例如，`192.168.1.xxx`）。
4. 保存並重新啟動 Shadowsocks-NG：
   - 點擊 “確定” 以保存更改。
   - 重新啟動 Shadowsocks-NG 客戶端以應用新設置。

對於 Clash：

1. 編輯配置文件：
   - 確保在您的 `config.yaml` 中啟用了 `allow-lan: true` 設置。
2. 綁定到所有接口：
   - 在配置中，設置 `allow-lan: true` 通常會將代理綁定到所有可用接口，包括 LAN。
3. 重新啟動 Clash：
   - 重新啟動 Clash 客戶端以應用更改。

---

3. 為您的 Mac 分配靜態 IP

為了確保 OpenWRT 路由器和 Mac 之間的連接穩定，請為您的 Mac 分配一個靜態 IP。

在 macOS 上分配靜態 IP 的步驟：

1. 打開系統偏好設置：
   - 點擊 Apple 菜單並選擇 “系統偏好設置”。
2. 導航到網絡設置：
   - 點擊 “網絡”。
3. 選擇您的活動連接：
   - 從左側邊欄中選擇 “Wi-Fi” 或 “以太網”，具體取決於您的 Mac 如何連接到路由器。
4. 配置 IPv4 設置：
   - 點擊 “高級...”。
   - 轉到 “TCP/IP” 選項卡。
   - 將 “配置 IPv4” 從 “使用 DHCP” 更改為 “手動”。
5. 設置靜態 IP 地址：
   - IP 地址：選擇一個在路由器 DHCP 範圍之外的 IP 以防止衝突（例如，`192.168.1.xxx`）。
   - 子網掩碼：通常為 `255.255.255.0`。
   - 路由器：您的路由器 IP 地址（例如，`192.168.1.1`）。
   - DNS 服務器：您可以使用路由器的 IP 或其他 DNS 服務，如 `8.8.8.8`。

6. 應用設置：
    - 點擊 “確定” ，然後點擊 “應用” 以保存更改。

---

4. 在 OpenWRT 上安裝和配置 Redsocks

Redsocks 是一個透明的 socks 重定向器，允許您通過 SOCKS5 代理路由網絡流量。我們將使用 Redsocks 將 OpenWRT 的流量重定向通過在您的 Mac 上運行的 Shadowsocks 代理。

步驟 1：安裝 Redsocks

1. 更新軟件包列表：

```
ssh root@<router_ip>
opkg update
```

2. 安裝 Redsocks：

```
opkg install redsocks
```

如果 Redsocks 在您的 OpenWRT 存儲庫中不可用，您可能需要手動編譯或使用替代軟件包。

步驟 2：配置 Redsocks

1. 創建或編輯 Redsocks 配置文件：

```
vi /etc/redsocks.conf
```

2. 添加以下配置：

```
base {
    log_debug = on;
    log_info = on;
    log = "file:/var/log/redsocks.log";
    daemon = on;
    redirector = iptables;
}
```

```
redsocks {
    local_ip = 0.0.0.0;
    local_port = 12345;  # Redsocks監聽的本地端口
    ip = xxx.xxx.xxx.xxx;  # Mac的靜態IP
    port = xxxxx;          # Shadowsocks-NG的本地SOCKS5代理端口
    type = socks5;
    login = "";           # 如果您的代理需要身份驗證
    password = "";
}
```

注意：

- `local_port`：Redsocks 監聽來自 iptables 重定向的傳入連接的端口。
- `ip` 和 `port`：指向您的 Mac 的 Shadowsocks SOCKS5 代理（基於之前的步驟的 `xxx.xxx.xxx.xxx:xxxxx`）。
- `type`：設置為 `socks5`，因為 Shadowsocks 提供 SOCKS5 代理。

3. 保存並退出：
   - 按 `ESC`，輸入`:wq`，然後按 `Enter`。
4. 創建日誌文件：

```
touch /var/log/redsocks.log
chmod 644 /var/log/redsocks.log
```

步驟 3：啟動 Redsocks 服務

1. 啟用 Redsocks 在啟動時啟動：

```
/etc/init.d/redsocks enable
```

2. 啟動 Redsocks：

```
/etc/init.d/redsocks start
```

3. 驗證 Redsocks 是否運行：

```
ps | grep redsocks
```

您應該看到一個 Redsocks 進程正在運行。

---

5. 將 OpenWRT 的流量重定向通過 macOS 代理

現在 Redsocks 已在 OpenWRT 上設置完畢，配置 iptables 以將所有出站 TCP 流量重定向通過 Redsocks，Redsocks 再將其路由通過您的 Mac 的 Shadowsocks 代理。

步驟 1：配置 iptables 規則

1. 添加 iptables 規則以重定向流量：

   ```
   # 將所有 TCP 流量重定向到 Redsocks（除了代理本身的流量）
   iptables -t nat -N REDSOCKS
   iptables -t nat -A REDSOCKS -d xxx.xxx.xxx.xxx -p tcp --dport xxxxx -j RETURN
   iptables -t nat -A REDSOCKS -p tcp -j REDIRECT --to-ports 12345

   # 將 REDSOCKS 鏈應用於所有出站流量
   iptables -t nat -A OUTPUT -p tcp -j REDSOCKS
   iptables -t nat -A PREROUTING -p tcp -j REDSOCKS
   ```

   解釋：

   - 創建新鏈：`REDSOCKS`
   - 排除代理流量：確保目的地為代理本身的流量不被重定向。
   - 重定向其他 TCP 流量：將其他 TCP 流量轉發到 Redsocks 的監聽端口（`12345`）。

2. 保存 iptables 規則：

   要使這些規則在重啟後持久化，請將它們添加到防火牆配置中。

   ```
   vi /etc/firewall.user
   ```

   添加 iptables 規則：

   ```
   # 將所有 TCP 流量重定向到 Redsocks（除了代理）
   iptables -t nat -N REDSOCKS
   iptables -t nat -A REDSOCKS -d xxx.xxx.xxx.xxx -p tcp --dport xxxxx -j RETURN
   iptables -t nat -A REDSOCKS -p tcp -j REDIRECT --to-ports 12345

   # 應用 REDSOCKS 鏈
   iptables -t nat -A OUTPUT -p tcp -j REDSOCKS
   iptables -t nat -A PREROUTING -p tcp -j REDSOCKS
   ```

保存並退出：

- 按 ESC，輸入:wq，然後按 Enter。

3. 重新啟動防火牆以應用更改：

```
/etc/init.d/firewall restart
```

步驟 2：驗證流量是否被重定向

1. 檢查 Redsocks 日誌：

```
cat /var/log/redsocks.log
```

您應該看到日誌指示流量正在通過 Redsocks 處理。

2. 從客戶端設備測試：

- 將設備連接到您的 OpenWRT 路由器。
- 訪問網站或執行使用互聯網的操作。
- 通過檢查外部 IP 地址（例如，通過WhatIsMyIP.com）驗證流量是否通過 Shadowsocks 代理路由，以查看是否反映代理的 IP。

---

6. 測試代理設置

通過執行以下測試確保整個設置按預期工作。

步驟 1：在 Mac 上驗證 Shadowsocks 連接

1. 檢查 Shadowsocks 客戶端狀態：

- 確保 Shadowsocks-NG 或 Clash 已主動連接到 Shadowsocks 服務器。
- 驗證本地代理（例如，xxx.xxx.xxx.xxx:xxxxx）是否可訪問。

2. 本地測試代理：

- 在您的 Mac 上，打開瀏覽器並將其配置為使用 localhost:1080 作為 SOCKS5 代理。
- 訪問WhatIsMyIP.com以確認 IP 是否匹配 Shadowsocks 服務器。

步驟 2：驗證 OpenWRT 的流量是否通過代理路由

1. 檢查 OpenWRT 的外部 IP：
   - 從連接到 OpenWRT 的設備，訪問WhatIsMyIP.com以查看 IP 是否反映 Shadowsocks 服務器的 IP。
2. 監控 Redsocks 日誌：
   - 在 OpenWRT 上，監控 Redsocks 日誌以確保流量正在被重定向。

```
tail -f /var/log/redsocks.log
```

3. 如有必要進行故障排除：
   - 如果流量未正確路由：
     - 確保 Mac 上的 Shadowsocks 客戶端正在運行且可訪問。
     - 驗證 iptables 規則是否正確設置。
     - 檢查 Mac 和 OpenWRT 上的防火牆設置。

其他注意事項

1. 安全性

- 保護您的代理：
  - 確保只有受信任的設備可以訪問代理。由於您正在通過 Redsocks 重定向所有流量，請確保 Mac 的防火牆僅允許來自 OpenWRT 路由器的連接。

  在 macOS 上：

  - 轉到系統偏好設置 > 安全與隱私 > 防火牆。
  - 配置防火牆以僅允許來自 OpenWRT 路由器 IP 的代理端口（`xxxxx`）的傳入連接。
- 身份驗證：
  - Shadowsocks 已經通過加密提供了一定程度的安全性。確保使用強密碼和加密方法。

2. 性能

- 路由器資源：
  - 運行 Redsocks 等代理服務可能會消耗 OpenWRT 路由器上的額外 CPU 和內存。確保您的路由器有足夠的資源。
- Mac 性能：
  - 確保您的 Mac 保持開機並連接到網絡以維持代理可用性。

3. 維護

- 監控日誌：
  - 定期檢查 Redsocks 和 Shadowsocks 日誌以查找任何異常活動或錯誤。
- 更新軟件：
  - 保持 OpenWRT、Redsocks 和您的 Shadowsocks 客戶端更新，以受益於安全補丁和性能改進。

4. 替代方法

雖然使用 Mac 作為中介代理服務器是可行的，但考慮以下替代方案以實現更簡單的設置：

- 直接在 OpenWRT 上配置 Shadowsocks 客戶端：
  - OpenWRT 通過 `shadowsocks-libev` 等軟件包直接支持 Shadowsocks。此方法消除了對 Mac 中介的需求。
- 使用專用代理/VPN 設備：
  - 像 Raspberry Pi 這樣的設備可以運行代理服務並充當專用網關。

結論

通過按照上述步驟操作，您已將 Mac 設置為 Shadowsocks 代理服務器，並配置了 OpenWRT 路由器以將所有連接設備的流量通過此代理路由。此設置通過利用 Shadowsocks 協議增強了網絡的隱私和控制。

關鍵點回顧：

1. Mac 上的 Shadowsocks 客戶端：
    - 安裝並配置了 Shadowsocks-NG 或 Clash 與您提供的 Shadowsocks URL。
    - 配置客戶端以監聽 Mac 的 LAN IP。
2. 代理可訪問性：
    - 為 Mac 分配了靜態 IP 以確保一致的代理訪問。
    - 配置了 macOS 防火牆以允許傳入代理連接。
3. OpenWRT 配置：
    - 安裝並配置了 Redsocks 以將所有出站 TCP 流量重定向通過 Shadowsocks 代理。
    - 應用了必要的 iptables 規則以強制流量重定向。
4. 測試：
    - 通過檢查外部 IP 地址驗證了來自連接設備的流量