📁 LPIC-2 / 🔒 Tema 212 - La seguridad del sistema

7.6.5 212.5 OpenVPN (peso: 2)

Teoría

Una VPN (Virtual Private Network) crea un "túnel" cifrado a través de una red pública (como Internet), permitiendo que los datos viajen de forma segura entre dos puntos (cliente y servidor, o dos redes). **OpenVPN** es una popular aplicación VPN de código abierto que implementa técnicas de red privada virtual para crear conexiones seguras punto a punto o sitio a sitio.

Características Clave de OpenVPN:

- Utiliza SSL/TLS para cifrar el túnel de datos y autenticar a los participantes.
- Puede operar sobre TCP o UDP. UDP suele ser más eficiente.
- Es muy configurable y soporta varios métodos de autenticación (certificados, clave pre-compartida, autenticación por nombre de usuario/contraseña con PAM).
- Crea interfaces de red virtuales (típicamente `tun` para enrutamiento de paquetes IP, `tap` para bridging a nivel de Ethernet).

Componentes de una Configuración OpenVPN Basada en Certificados:

- **Servidor OpenVPN:** El sistema que acepta las conexiones entrantes y actúa como puerta de enlace a la red privada.
- **Cliente(s) OpenVPN:** Los sistemas que inician la conexión al servidor.
- **Autoridad de Certificación (CA):** Una entidad de confianza (en este caso, creada por ti mismo) que emite y firma los certificados del servidor y los clientes. (Revisado 208.4 sobre CAs).
- **Certificado y Clave Privada del Servidor:** Identifican y autentican al servidor ante los clientes. Firmados por la CA.
- **Certificado y Clave Privada de cada Cliente:** Identifican y autentican a cada cliente ante el servidor. Firmados por la CA.
- **Certificado de la CA:** Necesario en el servidor y en cada cliente para verificar la autenticidad de los certificados de la otra parte.
- **Parámetros Diffie-Hellman (DH):** Utilizados en el proceso de intercambio seguro de claves.

Gestión de Certificados y Claves (Usando `easy-rsa`):

Crear la estructura de CA y los certificados/claves manualmente con `openssl` es complejo. La herramienta **`easy-rsa`** es un conjunto de scripts que simplifican este proceso. A menudo se instala como un paquete separado (`easy-rsa`).

- **Proceso General con `easy-rsa`:**
    1. Inicializar la CA (crear clave y certificado raíz).

2. Generar la solicitud de certificado (CSR) y clave privada para el servidor.
3. Firmar la solicitud del servidor con la clave de la CA.
4. Generar la solicitud de certificado y clave privada para cada cliente.
5. Firmar las solicitudes de cada cliente con la clave de la CA.
6. Generar los parámetros Diffie-Hellman (`dhparams.pem`).
7. (Opcional) Generar una clave de autenticación TLS pre-compartida (`ta.key`).

Configuración del Servidor OpenVPN:

1. **Instalación:**
    - **Paquete:** `openvpn` (estándar).
2. **Gestión del Servicio:**
    - **Nombre del Servicio (Diferencias):** A menudo se usa el modo de instancia de systemd: `openvpn@<nombre_config>.service`. El nombre de la instancia es el nombre del archivo de configuración (sin la extensión `.conf`) en `/etc/openvpn/`. Ej: si el archivo es `/etc/openvpn/server.conf`, el servicio es `openvpn@server.service`. También puede haber un servicio `openvpn.service` que arranca todas las configuraciones encontradas o una específica.
    - **Comandos Systemd:** `sudo systemctl enable openvpn@<nombre_config>`, `sudo systemctl start openvpn@<nombre_config>`.
3. **Archivo de Configuración:**
    - **Ubicación:** `/etc/openvpn/`. Cada archivo `.conf` es una configuración separada.
4. **Directivas Clave en el Archivo de Configuración del Servidor (`server.conf`):**
    - `port 1194`: Puerto en el que escuchar (1194 es el por defecto).
    - `proto udp`: Protocolo (udp o tcp).
    - `dev tun`: Tipo de interfaz virtual (tun para enrutamiento, tap para bridging).
    - `ca /etc/openvpn/certs/ca.crt`: Ruta al certificado de la CA.
    - `cert /etc/openvpn/certs/server.crt`: Ruta al certificado del servidor.
    - `key /etc/openvpn/certs/server.key`: Ruta a la clave privada del servidor.
    - `dh /etc/openvpn/certs/dhparams.pem`: Ruta a los parámetros Diffie-Hellman.
    - `server 10.8.0.0 255.255.255.0`: Define la subred IP virtual para la VPN (los clientes obtendrán IPs en este rango). OpenVPN configura automáticamente el enrutamiento para esta subred.
    - `ifconfig-pool-linear`: Asigna IPs linealmente del pool (`server` directive).

- `push "route 192.168.1.0 255.255.255.0"`: Envía una ruta a los clientes para que puedan acceder a la red privada detrás del servidor (ej: tu red LAN 192.168.1.0/24).
- `push "redirect-gateway def1"`: Envía una ruta a los clientes para que todo su tráfico de Internet pase por la VPN (el servidor OpenVPN se convierte en su gateway por defecto). Requiere NAT/Masquerading en el servidor.
- `keepalive 10 120`: Enviar pings cada 10 segundos, si no hay respuesta en 120 segundos, considera la conexión muerta.
- `cipher AES-256-CBC`: Algoritmo de cifrado. Elige uno seguro.
- `auth SHA256`: Algoritmo de autenticación para paquetes de datos.
- `user nobody`, `group nogroup`: Ejecutar el demonio con privilegios mínimos.
- `persist-key`, `persist-tun`: Evitar que la clave privada y la interfaz tun/tap se vuelvan a leer/configurar en caso de reinicios temporales.
- `status /var/log/openvpn/openvpn-status.log`: Archivo de estado de las conexiones activas.
- `log /var/log/openvpn/openvpn.log`: Archivo de log de OpenVPN.
- `verb 3`: Nivel de detalle del log (3 es un buen nivel para depuración).
- `client-config-dir /etc/openvpn/ccd`: Directorio opcional para configuraciones específicas por cliente (basado en el nombre común del certificado del cliente).

5. **Reenvío de IP y NAT:**
    - Si los clientes necesitan acceder a redes más allá del servidor OpenVPN (incluida Internet si usas `redirect-gateway`), debes habilitar el reenvío de IP en el servidor (Revisado 212.1).
    - Si los clientes necesitan acceder a Internet a través del servidor VPN (NAT/Masquerading), debes configurar reglas de firewall en el servidor para aplicar NAT al tráfico proveniente de la interfaz virtual OpenVPN (`tun0` o similar) saliendo por la interfaz pública (Revisado 212.1, 212.2).
6. **Firewall:**
    - El firewall del servidor OpenVPN debe permitir el tráfico entrante al puerto OpenVPN configurado (por defecto 1194 UDP/TCP).
    - Debes configurar el firewall para permitir o denegar el tráfico *enrutado* entre la interfaz virtual OpenVPN (`tun0`) y otras interfaces (LAN, WAN) en la cadena `FORWARD` (Revisado 212.1, 212.2).

Configuración del Cliente OpenVPN:

1. **Instalación:** Paquete `openvpn` (estándar).
2. **Archivo de Configuración:** Un archivo `.conf` o `.ovpn` que contiene las directivas del cliente. Puede distribuirse a los usuarios.

3. **Directivas Clave en el Archivo de Configuración del Cliente (`client.conf` o `.ovpn`):**
    - `client`: Indica que es un cliente.
    - `dev tun`: Tipo de interfaz virtual. Debe coincidir con el servidor.
    - `proto udp`: Protocolo. Debe coincidir con el servidor.
    - `remote <ip_o_nombre_servidor> <puerto>`: Dirección y puerto del servidor OpenVPN.
    - `ca <ruta_ca.crt>`: Ruta al certificado de la CA.
    - `cert <ruta_client.crt>`: Ruta al certificado del cliente.
    - `key <ruta_client.key>`: Ruta a la clave privada del cliente.
    - `resolv-retry infinite`: Reintentar indefinidamente la resolución de DNS del servidor.
    - `nobind`: No vincular a un puerto local específico (el kernel elige uno).
    - `persist-key`, `persist-tun`: Similares a las del servidor.
    - `remote-cert-tls server`: Opcional, verifica que el certificado del servidor es para un servidor TLS.
    - `key-direction 1`: Si se usa autenticación TLS (`tls-auth`), especifica la dirección (1 para cliente).
    - `verb 3`: Nivel de log.
4. **Autenticación TLS (`tls-auth` y `ta.key`):**
    - Una clave pre-compartida adicional (`ta.key`) usada para autenticar el canal de control TLS. Proporciona una capa adicional de defensa contra ataques de denegación de servicio y inundaciones UDP.
    - Se genera con `easy-rsa`. El archivo `ta.key` debe estar presente en el servidor y en todos los clientes.
    - En el servidor, se añade la directiva `tls-auth ta.key 0`. En el cliente, `tls-auth ta.key 1`.