ACTIVIDAD 2 - WIREGUARD MÉTODOS DE AUTENTICACIÓN

Cristóbal Suárez Abad

SEGURIDAD Y ALTA DISPONIBILIDAD - 2º ASIR

Índice

1. Identifica el método de autenticación de WireGuard que tienes configurado actualmente. Explica cómo funciona y sus características.

El método que tengo actualmente, el cual configuré en la práctica anterior, es el **método de criptografía de cifrado asimétrico**.

Funciona mediante el uso de claves públicas y privadas para comprobar la autenticidad de las credenciales de cada una de las partes.

En nuestro caso, se generaron claves para **WireGuard (wg0)** y el **Cliente (nosotros):**

Servidor:

- Clave Privada: eBuXARj+KzrYmqq9LD1MdbZ6/p/Cyh82YpOJl0tkkUA= (Secreta)
- Clave Pública: HguDn7fQKaXvUge8Vvtsgql+iLka5SFHWJvYtbmWdyE= (Compartida)

Cliente 1:

- Clave Privada: 4G0d8t1cl3uZQgfBRShg7pDx0RjYB2rgoYeKM4yKjWc= (Secreta)
- Clave Pública: 20ZjNZIasi7V6jS279wH2f1Q7FuI/KdR7HJVtiZp4ys= (Compartida),

Ambas partes comparten su clave pública en los respectivos archivos:

- En el lado del servidor, en “wg0.conf”, este tiene la su clave privada y la clave pública del cliente.
- En el lado del cliente, en “client1.conf” este tiene su clave privada la pública del servidor de WireGuard.

A la hora de conectarnos desde el Cliente al Servidor, ambos comprueban la identidad del otro usando estos ficheros.

2. Explica si existe algún método de autenticación nativo de WireGuard. Configúralo, explica cómo funciona y sus características.

Según la propia página de WireGuard[1], el único método de autenticación nativo es el uso de pares de claves públicas/privadas basado en el protocolo **Noise_IK Handshake[2]**.

Si bien es cierto que se indica el uso de un sistema para cuando es necesario realizar el método de encriptación simétrica, usando una clave pre-compartida (PSK) que se envía dentro junto con la clave pública y se encuentra en el documento de configuración del cliente[3].

Plantilla de ejemplo de archivo de configuración para el usuario:

[Interface]
PrivateKey = <Client private key>
 # Switch DNS server while connected.
 # Could be your internal DNS server, used on Omnia, or external
DNS = <your_server_subnet_IP> # to avoid DNS leaks

 # The addresses the client will bind to. Either IPv4 or IPv6.
 # Make sure to specify individual IPs for remote peers that don't
 # relay traffic and only act as simple clients (/32).
Address = 10.0.10.1/32

[Peer]
PublicKey = <Server public key>
 # Optional key known to both client and server; improves security
***PresharedKey** = <Pre-shared key from server for this client>*

 # The IP range that we may send packets to for this peer.
 # 0.0.0.0/0 will route all traffic through VPN
AllowedIPs = 0.0.0.0/0

 # Address of the server
Endpoint = <server IP>:<server port>

 # Send periodic keepalives to ensure connection stays up behind NAT.
PersistentKeepalive = 25

---

[1] https://www.wireguard.com/protocol/
[2] https://noiseprotocol.org/noise.pdf
[3] https://wiki.turris.cz/en/public/wireguard