Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Implementazione di una VPN di Management con WireGuard per il Controllo Remoto di Router OpenWRT via Ansible per reti IOT

Antonio Francesco Gentile[1], Davide Macrì[2],Emilio Greco [3]

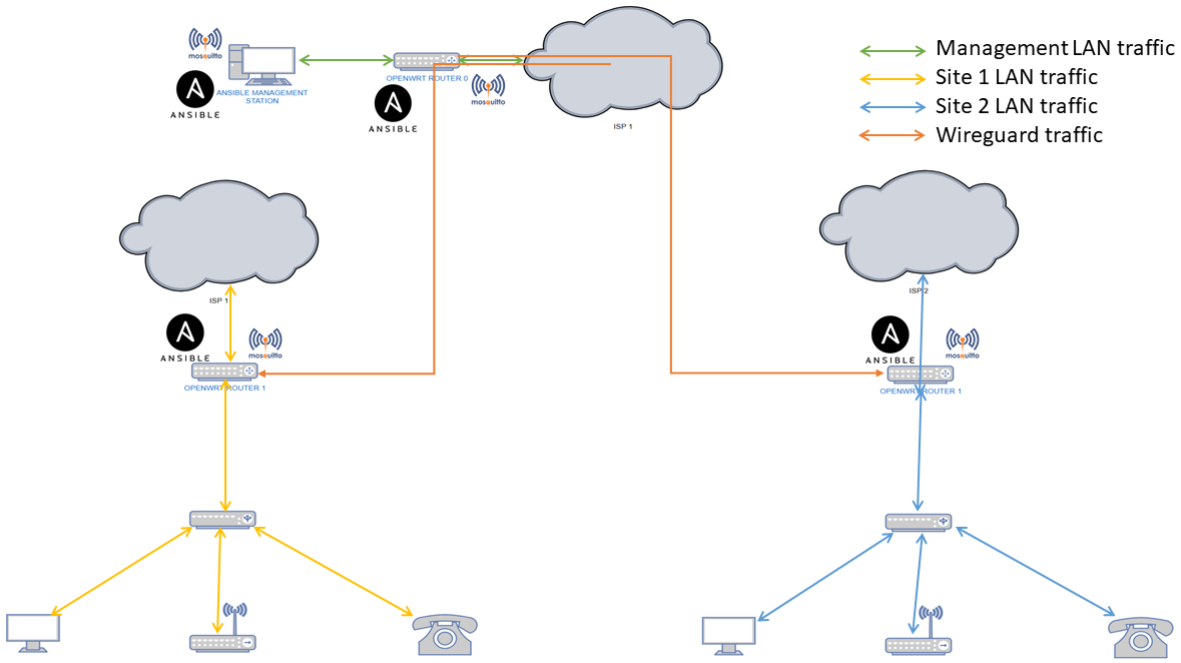
RT-ICAR-CS-24-04 **Ottobre 2024**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 8-9C, 87036 Rende, Italy, URL: www.icar.cnr.it
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.icar.cnr.it
– Sezione di Palermo, Via Ugo La Malfa, 153, 90146 Palermo, URL: www.icar.cnr.it

Introduzione

Questo documento descrive l'implementazione di una VPN di management basata su WireGuard per il controllo remoto di router OpenWRT. La soluzione proposta garantisce sicurezza, affidabilità e semplicità nella gestione dei router distribuiti geograficamente. Sono inclusi dettagli sulla configurazione di WireGuard, l'uso di Ansible per la gestione dei router e la configurazione di un broker MQTT (Mosquitto) per il monitoraggio degli stati delle operazioni.

La figura seguente illustra quanto realizzato:



Obiettivi del Progetto

1. **Sicurezza**: Implementare una connessione VPN sicura per gestire i router OpenWRT.
2. **Accessibilità**: Consentire il controllo remoto dei router tramite una singola interfaccia.
3. **Monitoraggio**: Raccogliere e gestire gli stati delle operazioni Ansible eseguite sui router.
4. **Automazione**: Utilizzare Ansible per automatizzare la configurazione e la gestione delle operazioni.

Architettura della Soluzione

La soluzione architettonica è composta da:

- **VPN WireGuard**: Per la connessione sicura tra il server di gestione e i router remoti.
- **Router OpenWRT**: Dispositivi di rete remoti da gestire.
- **Ansible**: Strumento di automazione per la gestione della configurazione dei router.
- **Mosquitto**: Broker MQTT per il monitoraggio e la comunicazione degli stati.

Fasi di Implementazione

1. Installazione di WireGuard sui Router OpenWRT

Passo 1: Installazione dei Pacchetti Necessari

Accedi a ciascun router OpenWRT e installa WireGuard:

```
opkg update
opkg install wireguard
opkg install mosquitto mosquitto-client libopenssl
```

Passo 2: Creazione delle Chiavi

Genera una chiave privata e una chiave pubblica per il router:

```
wg genkey | tee privatekey | wg pubkey > publickey
```

Passo 3: Configurazione di WireGuard

Crea un file di configurazione per WireGuard, ad esempio `/etc/config/wireguard`:

```
config wg0
    option private_key 'your_private_key'
    option listen_port '51820'
    list addresses '10.0.0.2/24'  # IP del router nella VPN
```

2. Configurazione del Server WireGuard

Passo 1: Installazione di WireGuard

Sul server di gestione, installa WireGuard:

```
apt update
apt install wireguard
```

Passo 2: Creazione delle Chiavi

Genera chiavi per il server:

```
wg genkey | tee server_privatekey | wg pubkey > server_publickey
```

Passo 3: Configurazione del Server

Configura il server WireGuard modificando il file `/etc/wireguard/wg0.conf`:

```
[Interface]
Address = 10.0.0.1/24  # IP del server nella VPN
PrivateKey = server_privatekey
ListenPort = 51820

[Peer]
PublicKey = router_public_key  # Chiave pubblica del router
AllowedIPs = 10.0.0.2/32
```

3. Avvio della VPN WireGuard

Passo 1: Avvio del Servizio

Avvia il server WireGuard:

```
wg-quick up wg0
```

Passo 2: Avvio del WireGuard sui Router

Avvia WireGuard sui router remoti:

```
wg-quick up wg0
```

4. Configurazione di Ansible per la Gestione dei Router

Passo 1: Installazione di Ansible

Installa Ansible sul server di gestione:

```
apt install ansible
```

Passo 2: Creazione dell'Inventario Ansible

Crea un file di inventario per i router OpenWRT:

```
[openwrt]
router1 ansible_host=10.0.0.2 ansible_user=root ansible_ssh_pass=password
router2 ansible_host=10.0.0.3 ansible_user=root ansible_ssh_pass=password
```

Passo 3: Esempio di Playbook Ansible

Ecco un esempio di playbook Ansible per configurare il firewall e pubblicare gli stati su Mosquitto:

```
---
- name: Configurazione Firewall e Monitoraggio
  hosts: openwrt
  tasks:
    - name: Inizializza stato
      set_fact:
        router_hostname: "{{ inventory_hostname }}"
        service_name: "firewall"
        service_state: "starting"

    - name: Pubblica stato iniziale su MQTT
      command: >
        python3 -c "import paho.mqtt.publish as publish;
        publish.single('/ansible/openwrt/{{ router_hostname }}/{{ service_name }}/{{
service_state }}',
        payload='Service is starting',
        hostname='localhost')"

    - name: Configura Firewall
      command: uci set firewall.@rule[0].enabled='1'
      register: firewall_config
```

```
    - name: Pubblica stato dopo configurazione del firewall
      command: >
        python3 -c "import paho.mqtt.publish as publish;
        publish.single('/ansible/openwrt/{{ router_hostname }}/{{ service_name
}}/configured',
        payload='Firewall has been configured',
        hostname='localhost')"
      when: firewall_config.changed

    - name: Riavvia il firewall
      command: /etc/init.d/firewall restart
      register: firewall_restart

    - name: Pubblica stato dopo riavvio del firewall
      command: >
        python3 -c "import paho.mqtt.publish as publish;
        publish.single('/ansible/openwrt/{{ router_hostname }}/{{ service_name
}}/running',
        payload='Firewall is running',
        hostname='localhost')"
      when: firewall_restart.changed

    - name: Imposta stato finale
      set_fact:
        service_state: "completed"

    - name: Pubblica stato finale su MQTT
      command: >
        python3 -c "import paho.mqtt.publish as publish;
        publish.single('/ansible/openwrt/{{ router_hostname }}/{{ service_name }}/{{
service_state }}',
        payload='Service completed successfully',
        hostname='localhost')"
```

5. Configurazione di Mosquitto per SSL e Autenticazione

Passo 1: Installazione di Mosquitto

Installa Mosquitto sul server:

```
apt install mosquitto mosquitto-clients
```

Passo 2: Creazione dei Certificati SSL

Genera chiavi e certificati per Mosquitto:

```
mkdir -p /etc/mosquitto/certs
openssl genrsa -out /etc/mosquitto/certs/mosquitto.key 2048
openssl req -new -x509 -key /etc/mosquitto/certs/mosquitto.key -out
/etc/mosquitto/certs/mosquitto.crt -days 365 -subj "/CN=your_domain_or_ip"
```

Passo 3: Creazione del File di Password

Crea il file di password per la gestione degli utenti:

```
mosquitto_passwd -c /etc/mosquitto/passwd your_username
```

Passo 4: Configurazione di Mosquitto

Configura Mosquitto per utilizzare SSL e autenticazione nel file */etc/mosquitto/mosquitto.conf:*

```
listener 8883
protocol mqtt

# SSL
cafile /etc/mosquitto/certs/mosquitto.crt
keyfile /etc/mosquitto/certs/mosquitto.key
certfile /etc/mosquitto/certs/mosquitto.crt

# Autenticazione
password_file /etc/mosquitto/passwd
allow_anonymous false
```

Passo 5: Riavvio di Mosquitto

Riavvia Mosquitto per applicare le modifiche:

```
systemctl restart mosquitto
```

6. Configurazione di Mosquitto Client SSL sui Router OpenWRT

1. Installazione di Mosquitto Client

Accedi a ciascun router OpenWRT e installa Mosquitto Client con supporto SSL:

```
opkg update
opkg install mosquitto-client libopenssl
```

2. Trasferimento dei Certificati SSL

Trasferisci i certificati dal server al router OpenWRT:

```
scp /path/to/certs/mosquitto.crt root@<router_ip>:/etc/mosquitto/certs/
```

3. Test della Connessione SSL

Testa la connessione al broker Mosquitto dal router OpenWRT:

```
mosquitto_sub -h your_broker_address -p 8883 -u your_username -P your_password -t
"/ansible/openwrt/#" --cafile /etc/mosquitto/certs/mosquitto.crt
```

7. Configurazione dei Router OpenWRT come Broker "Bridge"

1. Configurazione del File di Mosquitto

Modifica il file `/etc/mosquitto/mosquitto.conf` sui router OpenWRT per configurare il bridge:

```
# Configurazione del bridge
connection bridge_to_ubuntu
address your_ubuntu_ip:1883  # Indirizzo IP della macchina Ubuntu

topic # in 0
```

```
topic # out 0

# Se utilizzi l'autenticazione
remote_username your_username
remote_password your_password

# Configurazione delle opzioni di bridging
notifications true
```

2. Riavvio di Mosquitto

Riavvia Mosquitto sui router OpenWRT:

```
/etc/init.d/mosquitto restart
```

8. Monitoraggio degli Stati con MQTT

Configurare Mosquitto per pubblicare stati delle operazioni di Ansible usando il formato di topic:

```
/ansible/openwrt/routerHostname/servizio/stato
```

Dove `routerHostname` è il nome del router, `servizio` è il servizio monitorato e `stato` rappresenta lo stato attuale dell'operazione.

Ansible: Caratteristiche e Problemi Risolti

Ansible è uno strumento di automazione IT che consente di gestire configurazioni, implementazioni e orchestrazioni attraverso playbook scritti in YAML. Le sue caratteristiche principali includono:

- **Semplicità**: Utilizza un linguaggio di dichiarazione per configurare e gestire i sistemi.
- **Automazione**: Permette di automatizzare processi ripetitivi, riducendo il rischio di errore umano.
- **Flessibilità**: Supporta diversi sistemi operativi e applicazioni, facilitando la gestione di ambienti eterogenei.

Problemi Risolti da Ansible:

- **Configurazione Coerente**: Garantisce che tutti i dispositivi siano configurati in modo uniforme.
- **Riduzione dei Tempi di Configurazione**: Automatizza processi che richiederebbero molto tempo se eseguiti manualmente.
- **Versioning delle Configurazioni**: Permette di tenere traccia delle modifiche apportate alle configurazioni nel tempo.

WireGuard: Pro e Contro

WireGuard è una VPN moderna progettata per essere semplice, veloce e sicura. Di seguito sono riportati alcuni dei suoi vantaggi e svantaggi rispetto ad altre soluzioni VPN.

Vantaggi

1. **Semplicità di Configurazione**: WireGuard è più facile da configurare rispetto a soluzioni VPN più complesse come OpenVPN.
2. **Performance Elevate**: WireGuard offre prestazioni migliori grazie a un design più leggero e a meno overhead.
3. **Sicurezza**: Utilizza algoritmi di crittografia moderni e mantiene un codice sorgente ridotto, riducendo il rischio di vulnerabilità.

Svantaggi

1. **Maturità**: Sebbene WireGuard stia guadagnando popolarità, è relativamente nuovo e potrebbe non essere supportato da tutte le applicazioni o dispositivi.
2. **Funzionalità Limitate**: Rispetto a soluzioni più mature, potrebbe mancare alcune funzionalità avanzate come il routing a livello di applicazione.

Conclusioni

L'implementazione di una VPN di management utilizzando WireGuard per il controllo dei router OpenWRT, unita all'uso di Ansible per l'automazione e Mosquitto per il monitoraggio, offre un approccio robusto e scalabile per la gestione delle reti distribuite.

Questa architettura non solo migliora la sicurezza e l'affidabilità delle comunicazioni, ma semplifica anche la gestione dei dispositivi remoti, riducendo il carico di lavoro per gli amministratori di sistema. L'uso di Ansible consente di mantenere configurazioni coerenti e di eseguire operazioni in modo efficiente, mentre la comunicazione tramite MQTT offre un metodo efficace per monitorare le operazioni in tempo reale.

Considerazioni Finali

È fondamentale mantenere un monitoraggio costante della sicurezza della rete e aggiornare regolarmente le configurazioni di VPN e broker MQTT. La configurazione proposta può essere ulteriormente adattata e scalata in base alle necessità specifiche e alla crescita dell'infrastruttura di rete. In futuro, l'integrazione di strumenti di monitoraggio avanzati e di analytics potrebbe fornire ulteriori vantaggi operativi, ottimizzando ulteriormente la gestione delle risorse IT.