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

Deploy di reti Mesh via VPN con routing dinamico

Antonio Francesco Gentile, Davide Macrì, Emilio Greco

RT-ICAR-CS-25-02 **Gennaio 2025**

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

Indice generale

Introduzione

Questo rapporto tecnico illustra il processo di configurazione e implementazione di una rete mesh sicura composta da cinque nodi, interconnessi attraverso **VPN** e supportata da protocolli di routing dinamico. La progettazione di questa rete è stata sviluppata con l'obiettivo di garantire comunicazioni affidabili, scalabilità e sicurezza in ambienti che richiedono infrastrutture distribuite e resilienti.

Ogni nodo della rete è configurato per utilizzare una VPN, con una scelta tra **WireGuard**, **OpenVPN** o **Libreswan (IPsec)**, fornendo uno strato di protezione critico per la trasmissione dei dati. Parallelamente, i protocolli di routing dinamico, come **BGP** e **OSPF**, implementati attraverso il software **BIRD**, assicurano che i dati siano instradati in modo efficiente, anche in presenza di guasti o cambiamenti nella topologia della rete. L'architettura include anche configurazioni di firewall avanzate, realizzate utilizzando **iptables** e **nftables**, per proteggere i nodi da accessi non autorizzati e garantire un elevato livello di sicurezza.

Il lavoro si concentra sulla progettazione e configurazione di una rete mesh distribuita, rappresentata da una topologia a **cinque nodi connessi in modo ridondante**. Questo approccio consente di ottimizzare la resilienza della rete, garantendo che la comunicazione tra i nodi rimanga stabile anche in presenza di eventi imprevisti. L'uso delle VPN per la connessione tra i nodi protegge i dati durante il loro transito, mentre l'adozione dei protocolli di routing dinamico migliora l'efficienza e l'affidabilità della rete, rendendola capace di adattarsi a variazioni nei percorsi disponibili.

Il rapporto fornisce una descrizione dettagliata delle configurazioni richieste per ogni nodo della rete. Ogni nodo è configurato per comunicare tramite uno specifico protocollo VPN, selezionato in base alle esigenze dell'infrastruttura. WireGuard, OpenVPN e Libreswan rappresentano tecnologie chiave che garantiscono sicurezza e stabilità nella trasmissione dei dati. L'implementazione del routing dinamico è gestita attraverso configurazioni pratiche dei protocolli BGP e OSPF, che vengono spiegate nel dettaglio, evidenziando le scelte progettuali che assicurano la scalabilità e la flessibilità della rete.

Una sezione importante del documento è dedicata alla sicurezza. La configurazione dei firewall è trattata con grande attenzione, fornendo esempi concreti per proteggere la rete sia a livello di interfaccia che per i flussi di traffico interni. Le configurazioni di iptables e nftables rappresentano soluzioni affidabili e moderne per gestire le regole di sicurezza in ambienti complessi.

Questo rapporto è pensato per applicazioni pratiche in diversi contesti, tra cui organizzazioni distribuite che necessitano di una rete affidabile tra più sedi, ambienti critici come data center o infrastrutture di comunicazione e ambiti di ricerca e sviluppo per la simulazione e il testing di scenari complessi. L'approccio descritto garantisce alta disponibilità, sicurezza e flessibilità, rendendo questa soluzione ideale per affrontare le sfide di infrastrutture distribuite moderne.

Con questo documento, si fornisce una guida pratica per progettare e configurare reti mesh sicure, capaci di soddisfare esigenze elevate in termini di prestazioni e protezione dei dati.

Virtual Private Network

Una **VPN (Virtual Private Network)** è una tecnologia che consente di stabilire una connessione sicura e crittografata tra un dispositivo e una rete remota attraverso Internet. Questa soluzione è ampiamente utilizzata per garantire la privacy, la sicurezza dei dati e per permettere l'accesso a risorse remote o a contenuti con restrizioni geografiche. Utilizzando una VPN, il traffico dati viene incanalato attraverso un tunnel virtuale che protegge le informazioni trasmesse, rendendole inaccessibili a terzi, come hacker o provider di servizi Internet.

Il funzionamento di una VPN si basa su un server remoto gestito dal servizio VPN. Quando un utente si connette, il traffico Internet viene instradato attraverso questo server, che ne maschera l'indirizzo IP originale e lo sostituisce con uno proprio. Inoltre, tutti i dati scambiati vengono crittografati, garantendo che anche in caso di intercettazione, le informazioni rimangano indecifrabili.

Le VPN possono essere utilizzate per molteplici scopi, tra cui l'accesso sicuro a risorse aziendali, la protezione della privacy personale e la sicurezza durante l'utilizzo di reti Wi-Fi pubbliche. Sono strumenti indispensabili in scenari in cui la sicurezza delle comunicazioni è fondamentale, come negli ambienti aziendali o in ambiti in cui la censura e le restrizioni geografiche limitano l'accesso a Internet.

Tra le tecnologie che supportano le VPN, troviamo protocolli consolidati come **OpenVPN**, noto per la sua affidabilità e sicurezza, e **WireGuard**, che si distingue per un'implementazione più leggera e prestazioni elevate. Un altro protocollo comune è **IPsec**, ampiamente utilizzato per la sicurezza delle comunicazioni a livello IP.

L’utilizzo di una VPN offre numerosi benefici. Garantisce una protezione elevata dei dati, fondamentale per salvaguardare informazioni sensibili come credenziali o dati bancari. Inoltre, consente di nascondere l'indirizzo IP dell'utente, proteggendo la sua identità online e riducendo la possibilità di essere tracciato o monitorato. Per le aziende, una VPN rappresenta una soluzione per permettere ai dipendenti di accedere in sicurezza alle risorse interne da qualsiasi luogo, rendendola una componente essenziale per il lavoro remoto.

Tuttavia, è importante notare che la scelta di un servizio VPN affidabile è cruciale. La qualità della crittografia, la politica di conservazione dei dati e la giurisdizione del servizio possono influire significativamente sulla sicurezza e sulla privacy offerte. Inoltre, l'uso di una VPN può comportare una lieve riduzione della velocità di connessione, poiché i dati devono essere instradati attraverso server intermedi.

Routing Dinamico

Una componente fondamentale delle reti mesh, come quella descritta in questo documento, è l'adozione del **routing dinamico**. Il routing dinamico rappresenta un approccio avanzato alla gestione delle rotte di rete, in cui i percorsi tra i nodi vengono determinati e aggiornati automaticamente in base alle condizioni della rete. A differenza del routing statico, in cui le rotte sono predefinite e immutabili, il routing dinamico consente alla rete di adattarsi ai cambiamenti, come il guasto di un nodo o l’aggiunta di nuovi dispositivi, garantendo così una maggiore affidabilità e scalabilità.

Il routing dinamico si basa su protocolli specializzati, come **BGP (Border Gateway Protocol)** e **OSPF (Open Shortest Path First)**, che gestiscono la distribuzione e l'aggiornamento delle informazioni sulle rotte tra i nodi della rete. Questi protocolli sono essenziali per garantire che i pacchetti di dati raggiungano la loro destinazione nel modo più efficiente possibile.

Nel routing dinamico, i protocolli lavorano in tempo reale per monitorare lo stato dei nodi e delle connessioni. Quando un nodo diventa inattivo o una nuova connessione viene stabilita, i protocolli aggiornano automaticamente le tabelle di routing. Ciò consente alla rete di reagire rapidamente ai cambiamenti, evitando interruzioni o congestioni. Ad esempio, OSPF utilizza un algoritmo noto come "**Dijkstra's Shortest Path First**" per calcolare il percorso ottimale tra i nodi, mentre BGP gestisce le rotte a livello globale, stabilendo percorsi tra reti autonome.

Il routing dinamico offre numerosi vantaggi rispetto al routing statico. Innanzitutto, elimina la necessità di aggiornare manualmente le rotte, riducendo significativamente il carico amministrativo. Inoltre, migliora la resilienza della rete, poiché i protocolli dinamici possono deviare automaticamente il traffico in caso di guasti o congestioni. Questo approccio è particolarmente utile in reti complesse e distribuite, come le reti mesh descritte in questo rapporto.

Tra i protocolli più utilizzati troviamo BGP e OSPF. BGP è spesso utilizzato in reti globali per stabilire connessioni tra diversi sistemi autonomi, ed è ampiamente impiegato per la gestione delle rotte su Internet. La sua specifica è definita nella RFC 4271. OSPF, invece, è un protocollo interno che opera all'interno di una singola rete, garantendo la rapida convergenza delle rotte e l'efficienza del traffico. OSPF è ampiamente documentato e supportato da numerosi fornitori, tra cui Cisco.

La configurazione di questi protocolli richiede una pianificazione accurata e una comprensione approfondita delle necessità della rete. Per BGP, ad esempio, è essenziale definire chiaramente i prefissi IP e i numeri di sistema autonomo (AS), mentre per OSPF è necessario strutturare le aree di rete in modo logico, come illustrato in questa guida.

Applicazioni nelle Reti Mesh

Nelle reti mesh, il routing dinamico è essenziale per garantire una distribuzione efficiente del traffico tra i nodi. La topologia dinamica di queste reti, con connessioni ridondanti e percorsi multipli, si presta perfettamente all'utilizzo di protocolli di routing dinamico. BGP può essere utilizzato per gestire connessioni esterne o inter-network, mentre OSPF si occupa delle rotte interne alla rete.

Il routing dinamico rappresenta una soluzione indispensabile per le reti distribuite che richiedono flessibilità, resilienza e scalabilità. La sua implementazione, combinata con protocolli affidabili come BGP e OSPF, consente di ottimizzare le prestazioni della rete, garantendo al contempo la continuità operativa anche in scenari complessi.

Architettura di Rete

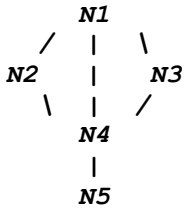
L'architettura della rete mesh proposta si basa su una configurazione composta da cinque nodi interconnessi, progettata per garantire resilienza, ridondanza e comunicazione sicura. Questa rete rappresenta un esempio pratico di infrastruttura distribuita in grado di adattarsi a scenari operativi complessi. La progettazione mira a ottimizzare la trasmissione dei dati tra i nodi, sfruttando sia le VPN come livello di sicurezza che i protocolli di routing dinamico per garantire l'efficienza nella gestione delle rotte.

Topologia della rete

La rete è rappresentata da una topologia a cinque nodi (N1, N2, N3, N4, N5), disposti in modo da formare una struttura mesh altamente connessa. Ogni nodo può comunicare direttamente con altri

nodi vicini, creando percorsi multipli e ridondanti. Questo approccio consente di evitare singoli punti di guasto e di mantenere la comunicazione anche in caso di malfunzionamenti di uno o più nodi.

La disposizione topologica può essere immaginata come segue:



Indirizzamento IP

Ogni nodo nella rete è configurato con un indirizzo IP univoco assegnato in base alla rete VPN. Gli indirizzi IP seguono uno schema numerico sequenziale per semplificare la configurazione e la gestione:

- Nodo 1 (N1): 10.0.0.1
- Nodo 2 (N2): 10.0.0.2
- Nodo 3 (N3): 10.0.0.3
- Nodo 4 (N4): 10.0.0.4
- Nodo 5 (N5): 10.0.0.5

Questi indirizzi IP operano all'interno della rete VPN, fornendo un'identità chiara e univoca a ciascun nodo. La configurazione garantisce che i dati possano essere instradati correttamente attraverso la rete, indipendentemente dal percorso scelto dai protocolli di routing.

Ogni nodo è configurato per stabilire connessioni sicure con i nodi adiacenti utilizzando uno dei protocolli VPN descritti nel rapporto (WireGuard, OpenVPN o Libreswan). La scelta del protocollo dipende dai requisiti specifici della rete, ma tutti garantiscono sicurezza crittografica e protezione dei dati in transito.

Ad esempio, il Nodo 1 (N1) è configurato per comunicare con i nodi N2, N3 e N4, creando tre percorsi indipendenti. Allo stesso modo, N4 è collegato a N1, N2, N3 e N5, fungendo da punto centrale per molteplici rotte. Questa struttura aumenta la tolleranza ai guasti, poiché eventuali disconnessioni o malfunzionamenti su un nodo possono essere aggirati tramite percorsi alternativi.

La progettazione di questa rete mesh garantisce un elevato grado di resilienza. In caso di malfunzionamento di un nodo, i protocolli di routing dinamico individuano automaticamente percorsi alternativi, evitando interruzioni nel flusso di dati. La ridondanza nelle connessioni tra i nodi minimizza l'impatto di eventuali problemi, garantendo continuità operativa.

Configurazione di OpenVPN

La configurazione di OpenVPN rappresenta **il primo passo nell'implementazione della rete mesh** descritta in questo documento. OpenVPN è una tecnologia VPN consolidata e ampiamente supportata, che offre flessibilità e robustezza nella creazione di tunnel sicuri per la trasmissione dei dati. Grazie alla sua compatibilità con una vasta gamma di dispositivi e sistemi operativi, OpenVPN costituisce una base affidabile per garantire la comunicazione sicura tra i nodi della rete.

In questa fase iniziale, OpenVPN viene utilizzato per stabilire una connessione stabile e funzionale tra i nodi, consentendo la costruzione di un'infrastruttura VPN essenziale per il funzionamento della rete mesh. La scelta di OpenVPN come tecnologia di riferimento iniziale permette di gestire configurazioni complesse, come l'autenticazione basata su certificati, e di affrontare in modo sistematico le sfide legate alla sicurezza e alla scalabilità. Questa configurazione getta le basi per integrare successivamente altre tecnologie VPN, garantendo una transizione fluida verso soluzioni più moderne, come WireGuard, o alternative compatibili, come Libreswan.

La rete VPN utilizza il protocollo UDP per la trasmissione dei dati, configurato su un'interfaccia virtuale TUN, che consente l'incapsulamento del traffico IP tra i nodi. Per iniziare, è necessario installare OpenVPN sul server designato. Su sistemi basati su Linux, come Ubuntu o CentOS, l'installazione può essere effettuata utilizzando i gestori di pacchetti predefiniti. Ad esempio, su Ubuntu, si può eseguire il comando `sudo apt-get install openvpn` per installare il pacchetto necessario. Una volta completata l'installazione, è essenziale configurare il server per gestire le connessioni dei client. Questo implica la creazione di una chiave e di un certificato per il server, nonché la configurazione dei parametri di rete appropriati.

Un passaggio cruciale nella configurazione di OpenVPN è l'istituzione di un'Autorità di Certificazione (CA). La CA è responsabile della firma dei certificati sia per il server che per i client, garantendo l'autenticità delle connessioni. Utilizzando strumenti come Easy-RSA, è possibile generare una CA privata, creare e firmare i certificati necessari. Questo processo assicura che solo i dispositivi autorizzati possano accedere alla VPN, mantenendo l'integrità e la sicurezza della rete.

Dopo aver creato i certificati, è necessario configurare il file di configurazione del server OpenVPN, solitamente denominato `server.conf`. Questo file definisce vari parametri, tra cui la porta su cui il server ascolta le connessioni in entrata (generalmente la porta UDP 1194), il protocollo da utilizzare (UDP o TCP), e le informazioni sui certificati e le chiavi. Inoltre, è possibile specificare le impostazioni per la rete virtuale, come l'intervallo di indirizzi IP da assegnare ai client connessi.

I client che desiderano connettersi al server OpenVPN devono avere il software OpenVPN installato e configurato correttamente. Ogni client necessita di un file di configurazione personalizzato, che include le informazioni sul server a cui connettersi, le chiavi e i certificati necessari per l'autenticazione. È fondamentale trasferire in modo sicuro i certificati e le chiavi dal server ai client, garantendo che solo dispositivi autorizzati possano stabilire una connessione.

La sicurezza è un aspetto cruciale nella configurazione di OpenVPN. È consigliabile utilizzare algoritmi di crittografia robusti e mantenere aggiornati sia il software del server che quello dei client per proteggersi da vulnerabilità note. Inoltre, l'implementazione di misure come l'autenticazione a due fattori può aggiungere un ulteriore livello di protezione, garantendo che solo utenti autorizzati possano accedere alla rete VPN.

Nodo 1 (N1) – Configurazione del Server OpenVPN

Il Nodo 1 (N1) è configurato come server OpenVPN, e il file di configurazione principale si trova in `/etc/openvpn/server.conf`. Questa configurazione definisce i parametri necessari per la gestione delle connessioni client e la sicurezza della rete.

```
port 1194
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh.pem
server 10.0.0.0 255.255.255.0
keepalive 10 120
cipher AES-256-CBC
persist-key
persist-tun
status /var/log/openvpn-status.log
verb 3
```

In questa configurazione, la porta 1194 è utilizzata per le connessioni in entrata, e il protocollo UDP è preferito per la sua efficienza in termini di velocità. I certificati e le chiavi di sicurezza sono generati e memorizzati nei percorsi specificati, assicurando che solo dispositivi autorizzati possano connettersi al server. La direttiva `server 10.0.0.0 255.255.255.0` definisce l'intervallo IP utilizzato per la rete virtuale, mentre `keepalive` specifica i parametri di monitoraggio per mantenere attive le connessioni client.

Nodo 2 (N2) – Configurazione del Client OpenVPN

Il Nodo 2 (N2) è configurato come client OpenVPN e utilizza il file `/etc/openvpn/client.conf` per connettersi al server (N1). La configurazione del client consente di stabilire una connessione sicura al server, utilizzando un'interfaccia TUN e certificati dedicati.

```
client
dev tun
proto udp
remote IP_NODE_1 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/client.crt
key /etc/openvpn/client.key
cipher AES-256-CBC
verb 3
```

Il parametro `remote IP_NODE_1 1194` specifica l'indirizzo IP del server (N1) e la porta utilizzata per le connessioni. I certificati e le chiavi del client, definiti nei percorsi `ca`, `cert` e `key`, devono essere univoci per ogni client per garantire autenticità e sicurezza. La configurazione include anche il parametro `resolv-retry infinite`, che consente al client di continuare a tentare la connessione in caso di fallimenti iniziali.

Nodi 3, 4 e 5 – Configurazione dei Client

I Nodi 3, 4 e 5 seguono una configurazione simile a quella del Nodo 2, adattata per utilizzare l'indirizzo IP del server OpenVPN (IP_NODE_1) e i propri certificati univoci. Ogni nodo è configurato per stabilire una connessione al server utilizzando la medesima rete VPN definita da N1. È fondamentale che ogni nodo abbia certificati e chiavi distinti per garantire una separazione delle identità nella rete.

Considerazioni sulla Sicurezza

Un aspetto critico nella configurazione di OpenVPN è la gestione dei certificati. La creazione di un'Autorità di Certificazione (CA) locale consente di firmare i certificati del server e dei client, garantendo che solo dispositivi autorizzati possano accedere alla rete VPN. Utilizzando algoritmi di crittografia robusti, come **AES-256-CBC**, si assicura che i dati trasmessi rimangano protetti da intercettazioni.

Inoltre, è consigliabile mantenere aggiornato il software OpenVPN su tutti i nodi per proteggere la rete da eventuali vulnerabilità. La configurazione del firewall sui nodi deve includere regole che limitino l'accesso alle porte utilizzate dal server VPN, consentendo solo connessioni da indirizzi IP autorizzati.

Dopo aver completato la configurazione, è importante testare la connessione VPN per assicurarsi che i client possano stabilire una comunicazione con il server e accedere alle risorse condivise. L'esecuzione di strumenti diagnostici, come `ping` e `traceroute`, può aiutare a identificare eventuali problemi di connettività.

Configurazione di WireGuard

La configurazione di WireGuard si colloca come una fase successiva all'implementazione di OpenVPN, offrendo una soluzione moderna e ottimizzata per migliorare le prestazioni della rete mesh. WireGuard è una tecnologia VPN progettata per essere leggera, veloce e altamente sicura, rappresentando una scelta eccellente per scenari in cui l'efficienza e la semplicità sono prioritarie.

L'integrazione di WireGuard dopo OpenVPN consente di sfruttare l'esperienza acquisita nella configurazione e gestione delle VPN per implementare una soluzione che riduca la complessità operativa, migliorando al contempo la velocità di trasmissione e l'efficienza del sistema. WireGuard, grazie alla sua architettura snella e al supporto di moderni algoritmi crittografici, rappresenta una naturale evoluzione per reti che richiedono flessibilità e scalabilità.

Questo capitolo approfondisce la configurazione di WireGuard su ciascun nodo della rete, fornendo indicazioni pratiche per integrare questa tecnologia con le configurazioni esistenti. Le impostazioni descritte sono progettate per garantire che WireGuard coesista armoniosamente con altre tecnologie VPN, come OpenVPN, permettendo una gestione fluida e sicura della rete mesh.

In questa sezione, viene descritta la configurazione di WireGuard per una rete mesh composta da cinque nodi. Ogni nodo viene configurato per comunicare con gli altri utilizzando una chiave privata univoca e una chiave pubblica condivisa con i peer. La rete VPN opera su un'interfaccia virtuale denominata **wg0**, e utilizza indirizzi IP privati assegnati staticamente a ciascun nodo.

Nodo 1 (N1) – Configurazione del Server WireGuard

Il file di configurazione per il Nodo 1 è situato in `/etc/wireguard/wg0.conf` e definisce le impostazioni dell'interfaccia WireGuard, inclusa la chiave privata, l'indirizzo IP della VPN, e i dettagli per la comunicazione con ciascun peer.

```
[Interface]
PrivateKey = PRIV_KEY_NODE_1
Address = 10.0.0.1/24
ListenPort = 51820

[Peer]
PublicKey = PUB_KEY_NODE_2
Endpoint = IP_NODE_2:51820
AllowedIPs = 10.0.0.2/32

[Peer]
PublicKey = PUB_KEY_NODE_3
Endpoint = IP_NODE_3:51820
AllowedIPs = 10.0.0.3/32

[Peer]
PublicKey = PUB_KEY_NODE_4
Endpoint = IP_NODE_4:51820
AllowedIPs = 10.0.0.4/32

[Peer]
PublicKey = PUB_KEY_NODE_5
Endpoint = IP_NODE_5:51820
AllowedIPs = 10.0.0.5/32
```

Nella configurazione, la sezione `[Interface]` contiene la chiave privata del Nodo 1 (PRIV_KEY_NODE_1), che non deve essere condivisa. L'indirizzo `10.0.0.1/24` è assegnato al nodo come indirizzo VPN. La porta 51820 è utilizzata per ascoltare le connessioni in entrata.

La sezione `[Peer]` specifica i dettagli per la comunicazione con gli altri nodi. Ogni peer è identificato da una chiave pubblica (ad esempio, PUB_KEY_NODE_2 per il Nodo 2) e dall'indirizzo IP e porta del peer remoto. Il parametro `AllowedIPs` limita il traffico indirizzato a ciascun peer all'intervallo di IP definito.

Nodo 2 (N2) – Configurazione del Client WireGuard

Il Nodo 2 utilizza una configurazione simile, definita in `/etc/wireguard/wg0.conf`. Anche in questo caso, la chiave privata è unica per il nodo, e le informazioni dei peer vengono configurate per consentire la comunicazione con gli altri nodi.

```
[Interface]
PrivateKey = PRIV_KEY_NODE_2
Address = 10.0.0.2/24
ListenPort = 51820

[Peer]
PublicKey = PUB_KEY_NODE_1
Endpoint = IP_NODE_1:51820
AllowedIPs = 10.0.0.1/32

[Peer]
```

```
PublicKey = PUB_KEY_NODE_3
Endpoint = IP_NODE_3:51820
AllowedIPs = 10.0.0.3/32

[Peer]
PublicKey = PUB_KEY_NODE_4
Endpoint = IP_NODE_4:51820
AllowedIPs = 10.0.0.4/32

[Peer]
PublicKey = PUB_KEY_NODE_5
Endpoint = IP_NODE_5:51820
AllowedIPs = 10.0.0.5/32
```

In questo file di configurazione, il Nodo 2 utilizza l'indirizzo IP `10.0.0.2/24`. Le chiavi pubbliche e gli endpoint per ciascun peer sono configurati per consentire comunicazioni sicure con i nodi rimanenti.

Nodi 3, 4 e 5 – Configurazione Simile

La configurazione per i Nodi 3, 4 e 5 segue lo stesso schema, con l'unica differenza negli indirizzi IP assegnati e nelle chiavi private utilizzate da ciascun nodo. Per esempio, il Nodo 3 utilizzerà un indirizzo IP `10.0.0.3/24` e una chiave privata unica (PRIV_KEY_NODE_3). Analogamente, le configurazioni dei peer includeranno i dettagli di tutti gli altri nodi della rete.

Ogni nodo utilizza una chiave pubblica e privata generata tramite gli strumenti forniti da WireGuard. La chiave privata viene mantenuta segreta sul nodo locale, mentre la chiave pubblica viene condivisa con i peer. Questo modello crittografico garantisce che solo i dispositivi autorizzati possano comunicare all'interno della rete VPN.

Per garantire la sicurezza, è essenziale:

1. Proteggere i file di configurazione da accessi non autorizzati.
2. Utilizzare regole di firewall per consentire solo il traffico sulle porte WireGuard (51820 in questo esempio).
3. Monitorare regolarmente la rete per rilevare anomalie o tentativi di accesso non autorizzati.

Una volta completata la configurazione, è fondamentale verificare che ogni nodo possa comunicare correttamente con gli altri. Questo può essere fatto utilizzando comandi come `ping` per testare la connettività VPN o `wg show` per visualizzare lo stato delle connessioni WireGuard.

Configurazione di Libreswan (IPsec)

La configurazione di **Libreswan** per la gestione delle connessioni IPsec rappresenta un passo fondamentale per garantire la sicurezza e l'integrità delle comunicazioni all'interno della rete mesh. IPsec (Internet Protocol Security) è una suite di protocolli progettata per proteggere le comunicazioni a livello di rete, offrendo funzionalità di crittografia e autenticazione dei pacchetti IP. Libreswan è una delle implementazioni più diffuse di IPsec su sistemi Linux, riconosciuta per la sua affidabilità e flessibilità nella gestione di connessioni VPN sicure.

In questo capitolo, ci concentreremo sulla configurazione di Libreswan per stabilire connessioni sicure tra i nodi della rete. Questo approccio è particolarmente utile in scenari che richiedono un elevato livello di sicurezza a livello di rete e compatibilità con standard consolidati. La configurazione

di IPsec con Libreswan consente di implementare politiche di sicurezza robuste, garantendo che i dati trasmessi tra i nodi siano protetti da intercettazioni e accessi non autorizzati.

La configurazione di Libreswan implica la definizione dei parametri necessari per stabilire connessioni sicure tra i nodi della rete. Questo processo coinvolge principalmente la modifica dei file di configurazione **`/etc/ipsec.conf`** e **`/etc/ipsec.secrets`**, che specificano rispettivamente i dettagli delle connessioni e le chiavi di autenticazione utilizzate per garantire la sicurezza delle comunicazioni.

Nodo 1 (N1) – Configurazione di IPsec

Per il Nodo 1 (N1), il file **`/etc/ipsec.conf`** deve essere configurato per definire le connessioni verso gli altri nodi della rete. Ogni connessione è identificata da un nome univoco e specifica gli indirizzi IP dei nodi coinvolti, le subnet interessate e il metodo di autenticazione utilizzato.

```
config setup
    protostack=netkey
    nat_traversal=yes
    virtual_private=%v4:10.0.0.0/24

conn N1-to-N2
    left=IP_NODE_1
    leftsubnet=10.0.0.1/32
    right=IP_NODE_2
    rightsubnet=10.0.0.2/32
    authby=secret
    auto=start

conn N1-to-N3
    left=IP_NODE_1
    leftsubnet=10.0.0.1/32
    right=IP_NODE_3
    rightsubnet=10.0.0.3/32
    authby=secret
    auto=start

conn N1-to-N4
    left=IP_NODE_1
    leftsubnet=10.0.0.1/32
    right=IP_NODE_4
    rightsubnet=10.0.0.4/32
    authby=secret
    auto=start

conn N1-to-N5
    left=IP_NODE_1
    leftsubnet=10.0.0.1/32
    right=IP_NODE_5
    rightsubnet=10.0.0.5/32
    authby=secret
    auto=start
```

In questa configurazione:

- **protostack=netkey**: specifica lo stack di protocollo da utilizzare (in questo caso, Netkey, lo stack IPsec predefinito per il kernel Linux).
- **nat_traversal=yes**: abilita il supporto per NAT Traversal, permettendo a IPsec di funzionare correttamente anche attraverso dispositivi NAT.

- **virtual_private=%v4:10.0.0.0/24**: definisce la rete privata virtuale utilizzata dai nodi.

Le sezioni **conn N1-to-NX** definiscono le connessioni tra il Nodo 1 e gli altri nodi (N2, N3, N4, N5), specificando gli indirizzi IP locali e remoti, le subnet e il metodo di autenticazione tramite chiave precondivisa (**authby=secret**).

Nodi 2, 3, 4 e 5 – Configurazione di IPsec

La configurazione per i Nodi 2, 3, 4 e 5 segue uno schema simile, con adeguamenti nei parametri **`left`** e **`right`** per riflettere gli indirizzi IP specifici di ciascun nodo. Ad esempio, per il Nodo 2 (N2), la configurazione nel file **`/etc/ipsec.conf`** potrebbe essere:

```
config setup
    protostack=netkey
    nat_traversal=yes
    virtual_private=%v4:10.0.0.0/24

conn N2-to-N1
    left=IP_NODE_2
    leftsubnet=10.0.0.2/32
    right=IP_NODE_1
    rightsubnet=10.0.0.1/32
    authby=secret
    auto=start

conn N2-to-N3
    left=IP_NODE_2
    leftsubnet=10.0.0.2/32
    right=IP_NODE_3
    rightsubnet=10.0.0.3/32
    authby=secret
    auto=start

conn N2-to-N4
    left=IP_NODE_2
    leftsubnet=10.0.0.2/32
    right=IP_NODE_4
    rightsubnet=10.0.0.4/32
    authby=secret
    auto=start

conn N2-to-N5
    left=IP_NODE_2
    leftsubnet=10.0.0.2/32
    right=IP_NODE_5
    rightsubnet=10.0.0.5/32
    authby=secret
    auto=start
```

È importante assicurarsi che ogni nodo abbia le configurazioni corrette per comunicare con gli altri nodi, specificando accuratamente gli indirizzi IP e le subnet coinvolte.

File delle Chiavi Segrete

Il file **`/etc/ipsec.secrets`** è utilizzato per memorizzare le chiavi segrete pre-condivise (PSK) che vengono utilizzate per l'autenticazione tra i nodi. La configurazione di questo file garantisce che ogni coppia di nodi possa autenticarsi correttamente durante la connessione.

```
IP_NODE_1 IP_NODE_2 : PSK "pre-shared-key"
IP_NODE_1 IP_NODE_3 : PSK "pre-shared-key"
IP_NODE_1 IP_NODE_4 : PSK "pre-shared-key"
IP_NODE_1 IP_NODE_5 : PSK "pre-shared-key"

IP_NODE_2 IP_NODE_1 : PSK "pre-shared-key"
IP_NODE_3 IP_NODE_1 : PSK "pre-shared-key"
IP_NODE_4 IP_NODE_1 : PSK "pre-shared-key"
IP_NODE_5 IP_NODE_1 : PSK "pre-shared-key"
```

In questo file, ogni coppia di nodi è associata a una chiave pre-condivisa (PSK) che viene utilizzata durante il processo di handshake per stabilire una connessione sicura. È fondamentale che ogni PSK sia mantenuta sicura e che non venga mai esposta in modo pubblico.

Una volta configurato Libreswan su tutti i nodi, è necessario eseguire dei test per assicurarsi che le connessioni siano stabilite correttamente. Si può utilizzare il comando ipsec status per verificare lo stato delle connessioni IPsec e diagnosticare eventuali problemi. È anche consigliabile utilizzare strumenti come ping e traceroute per testare la connettività tra i nodi attraverso la rete VPN.

Configurazione di BIRD per BGP

La rete mesh descritta in questo rapporto si basa su protocolli di routing dinamico per garantire una comunicazione efficiente, resiliente e scalabile tra i nodi. In questo contesto, la configurazione di BIRD per il protocollo BGP (Border Gateway Protocol) gioca un ruolo chiave nella gestione delle rotte e nella distribuzione del traffico tra i nodi della rete. La scelta di BGP consente ai nodi di comunicare in modo ottimale anche in presenza di cambiamenti nella topologia, garantendo che i dati siano sempre instradati lungo il percorso più appropriato.

In particolare, ogni nodo della rete è configurato per operare all'interno dello stesso sistema autonomo (AS) con numero **64512**, che identifica un gruppo di reti gestite sotto un'unica politica di routing. Questo approccio semplifica la configurazione e consente ai nodi di condividere informazioni sulle rotte disponibili, assicurando un funzionamento stabile anche in scenari complessi.

L'utilizzo di BIRD, un software leggero e altamente configurabile per il routing dinamico, permette di implementare con efficienza le sessioni BGP tra i nodi. BIRD è stato scelto per la sua compatibilità con reti distribuite e per le sue capacità avanzate nella gestione delle rotte e nel monitoraggio delle connessioni. Grazie alla sua flessibilità, BIRD si integra perfettamente con gli altri componenti della rete descritti nel rapporto, tra cui le VPN (OpenVPN, WireGuard e Libreswan) e le configurazioni di sicurezza dei firewall.

L'obiettivo di questa sezione è fornire una guida dettagliata per configurare BIRD su ciascun nodo, contestualizzandone l'implementazione all'interno della rete mesh. La configurazione garantisce che ogni nodo possa scambiare informazioni di routing con gli altri in modo sicuro e ottimizzato, massimizzando la resilienza della rete e minimizzando l'impatto di eventuali guasti.

Attraverso la configurazione di BIRD per BGP, la rete mesh acquisisce la capacità di adattarsi dinamicamente alle variazioni del traffico e alle modifiche della topologia, rendendola ideale per ambienti che richiedono alta disponibilità, scalabilità e robustezza.

Nodo 1 (N1) – Configurazione di BIRD

La configurazione di BIRD per il Nodo 1 (N1) è definita nel file `/etc/bird/bird.conf`. Questo file contiene tutte le direttive necessarie per configurare il nodo come router BGP. La configurazione include l'identificazione del nodo, i parametri per l'interazione con il kernel del sistema operativo e la definizione delle sessioni BGP con gli altri nodi della rete.

```
router id 10.0.0.1;

protocol kernel {
    scan time 10;
    import all;
    export all;
}

protocol device {
    scan time 10;
}

protocol bgp N1-to-N2 {
    local as 64512;
    neighbor 10.0.0.2 as 64512;
    import all;
    export all;
}

protocol bgp N1-to-N3 {
    local as 64512;
    neighbor 10.0.0.3 as 64512;
    import all;
    export all;
}

protocol bgp N1-to-N4 {
    local as 64512;
    neighbor 10.0.0.4 as 64512;
    import all;
    export all;
}

protocol bgp N1-to-N5 {
    local as 64512;
    neighbor 10.0.0.5 as 64512;
    import all;
    export all;
}
```

Dettagli della configurazione:

- **router id 10.0.0.1;**: Identifica univocamente il nodo all'interno del sistema autonomo. Ogni nodo della rete deve avere un ID router univoco.
- **protocol kernel**: Consente l'interazione tra BIRD e il kernel del sistema operativo, sincronizzando le tabelle di routing.
- **protocol device**: Configura il monitoraggio delle interfacce di rete per rilevare modifiche nello stato delle connessioni.
- **protocol bgp**: Definisce le sessioni BGP tra N1 e gli altri nodi (N2, N3, N4, N5). Ogni sessione specifica:
    - **local as 64512**: Il numero del sistema autonomo locale.

- **neighbor**: L'indirizzo IP e il sistema autonomo del nodo remoto.
- **import all; export all;**: Indica che tutte le rotte devono essere importate ed esportate.

Nodi 2, 3, 4 e 5 – Configurazioni Specifiche

Per i Nodi 2, 3, 4 e 5, le configurazioni seguono lo stesso schema del Nodo 1, con modifiche al parametro `router id` e agli indirizzi IP dei vicini. Ad esempio, per il Nodo 2 (N2), il file `/etc/bird/bird.conf` sarà:

```
router id 10.0.0.2;

protocol kernel {
    scan time 10;
    import all;
    export all;
}

protocol device {
    scan time 10;
}

protocol bgp N2-to-N1 {
    local as 64512;
    neighbor 10.0.0.1 as 64512;
    import all;
    export all;
}

protocol bgp N2-to-N3 {
    local as 64512;
    neighbor 10.0.0.3 as 64512;
    import all;
    export all;
}

protocol bgp N2-to-N4 {
    local as 64512;
    neighbor 10.0.0.4 as 64512;
    import all;
    export all;
}

protocol bgp N2-to-N5 {
    local as 64512;
    neighbor 10.0.0.5 as 64512;
    import all;
    export all;
}
```

Ogni nodo deve configurare il proprio **router id** e specificare le connessioni BGP con gli altri nodi, utilizzando gli indirizzi IP appropriati per i vicini.

Nella configurazione di BGP con BIRD, è importante adottare misure di sicurezza per proteggere le sessioni di routing:

1. **Autenticazione delle sessioni BGP**: Configurare una password (opzione MD5) per autenticare le connessioni tra i nodi e prevenire accessi non autorizzati.

2. **Filtraggio delle rotte**: Implementare regole per limitare le rotte importate o esportate, riducendo il rischio di configurazioni errate o attacchi di tipo route hijacking.
3. **Monitoraggio del traffico**: Utilizzare strumenti come Wireshark per analizzare il traffico BGP e individuare eventuali anomalie.

Dopo aver configurato BIRD su tutti i nodi, è necessario verificare che le sessioni BGP siano attive e funzionanti. Utilizzare il comando:

```
birdc show protocols
```

per visualizzare lo stato delle sessioni BGP. È importante controllare che le rotte siano propagate correttamente tra i nodi e che non ci siano errori di configurazione.

Configurazione di BIRD per OSPF

Nel contesto della rete mesh sicura descritta in questo rapporto, la configurazione di **BIRD** per il protocollo **OSPF (Open Shortest Path First)** riveste un ruolo centrale nell'implementazione di un routing dinamico interno efficiente. OSPF, come protocollo di routing di tipo link-state, è progettato per calcolare i percorsi ottimali basandosi su metriche definite, come il costo, e per adattarsi rapidamente alle variazioni della topologia. Questo lo rende particolarmente adatto per reti distribuite e resiliente come quelle basate su una topologia mesh.

Mentre il capitolo precedente ha affrontato la configurazione di BIRD per BGP, utilizzato per il routing tra sistemi autonomi (AS), l'implementazione di OSPF permette di ottimizzare il routing interno alla rete, migliorando la gestione delle rotte tra i nodi. OSPF garantisce che ogni nodo possa determinare il percorso più breve verso gli altri, sfruttando un modello distribuito per calcolare le rotte in modo indipendente ma coerente con gli altri nodi.

La configurazione di OSPF si integra con le altre tecnologie utilizzate nella rete mesh, come VPN (WireGuard, OpenVPN e Libreswan) e firewall (iptables, nftables), e utilizza BIRD come framework per implementare il protocollo in modo flessibile ed efficiente. La scelta di OSPF, rispetto ad altri protocolli di routing interno, deriva dalla sua capacità di scalare facilmente, dalla rapidità nella convergenza delle rotte e dalla possibilità di segmentare la rete in aree logiche per ottimizzare le risorse.

Questo capitolo fornisce una guida pratica alla configurazione di BIRD per OSPF, evidenziando come ogni nodo della rete può essere configurato per operare come router OSPF, scambiando informazioni sulle rotte con gli altri nodi in modo sicuro e affidabile.

Nodo 1 (N1) – Configurazione di OSPF

Il file di configurazione di BIRD per il Nodo 1 (N1), situato in `/etc/bird/bird.conf`, definisce i parametri per il routing dinamico basato su OSPF. La configurazione si concentra sull'identificazione del nodo, l'interazione con il kernel del sistema operativo, la gestione delle interfacce e la definizione delle aree OSPF.

```
router id 10.0.0.1;

protocol kernel {
    scan time 10;
    import all;
    export all;
```

```
}

protocol device {
    scan time 10;
}

protocol ospf {
    area 0.0.0.0 {
        interface "wg0" {
            type broadcast;
            cost 10;
        }
    }
}
```

Dettagli della configurazione:

- **router id 10.0.0.1;**: Identifica il nodo in modo univoco all'interno del dominio OSPF. Ogni nodo deve avere un router ID distinto.
- **protocol kernel**: Consente a BIRD di sincronizzare le rotte calcolate con le tabelle di routing del kernel.
- **protocol device**: Configura il monitoraggio delle interfacce fisiche e virtuali, come wg0.
- **protocol ospf**: Configura OSPF specificando un'unica area (0.0.0.0) che include l'interfaccia wg0.
    - **type broadcast**: Specifica che l'interfaccia opera in modalità broadcast, ideale per reti mesh.
    - **cost 10**: Definisce la metrica dell'interfaccia, utilizzata per calcolare il costo dei percorsi.

Nodi 2, 3, 4 e 5 – Configurazioni Specifiche

I file di configurazione per i Nodi 2, 3, 4 e 5 seguono lo stesso schema, con differenze nei parametri **`router id`** e nei dettagli dell'interfaccia. Ad esempio, per il Nodo 2 (N2), il file `/etc/bird/bird.conf` sarà:

```
router id 10.0.0.2;

protocol kernel {
    scan time 10;
    import all;
    export all;
}

protocol device {
    scan time 10;
}

protocol ospf {
    area 0.0.0.0 {
        interface "wg0" {
            type broadcast;
            cost 10;
        }
    }
}
```

Come per il Nodo 1, anche qui il **router id** deve essere unico, e l'interfaccia `wg0` rappresenta il punto di connessione principale per le comunicazioni OSPF.

L'implementazione di OSPF richiede un'attenta gestione della sicurezza per evitare configurazioni errate o attacchi alla rete. Tra le misure consigliate:

1. **Filtraggio delle interfacce**: Configurare regole firewall per limitare il traffico OSPF alle sole interfacce autorizzate.
2. **Autenticazione OSPF**: Implementare meccanismi di autenticazione, come password o chiavi, per proteggere gli scambi di messaggi OSPF.
3. **Monitoraggio delle connessioni**: Utilizzare strumenti diagnostici come birdc per verificare lo stato delle connessioni OSPF.

Dopo aver configurato BIRD per OSPF su tutti i nodi, è importante verificare che le sessioni siano attive e che le rotte siano propagate correttamente. Utilizzare i seguenti comandi:

```
birdc show protocols
birdc show ospf neighbors
```

Questi comandi forniscono informazioni sullo stato delle sessioni OSPF e sull'elenco dei vicini attivi, consentendo di diagnosticare eventuali problemi.

Configurazione del firewall

La configurazione del firewall rappresenta una parte essenziale della rete mesh sicura descritta in questo rapporto tecnico. Un firewall ben configurato garantisce che solo il traffico legittimo possa attraversare i nodi della rete, proteggendoli da accessi non autorizzati e da attacchi potenziali. In questa sezione, verranno presentate configurazioni basate su **iptables** e **nftables**, due strumenti potenti e ampiamente utilizzati per la gestione delle regole di filtraggio del traffico su sistemi Linux.

Nel contesto della rete mesh, il firewall ha il compito di:

1. Proteggere i nodi da traffico non autorizzato o potenzialmente dannoso.
2. Garantire che solo il traffico VPN (OpenVPN, WireGuard e Libreswan/IPsec) sia accettato sui nodi.
3. Consentire il transito del traffico legittimo tra i nodi della rete mesh (indirizzi IP 10.0.0.0/24).
4. Bloccare tutto il traffico non esplicitamente autorizzato, implementando una politica di **default deny**.

Le configurazioni fornite in questo capitolo possono essere applicate a ciascun nodo della rete, con modifiche minime necessarie per adattarsi agli specifici indirizzi e ruoli dei nodi.

Configurazione con iptables

iptables è uno strumento legacy ma ancora ampiamente utilizzato per la configurazione di firewall in ambienti Linux. Di seguito è riportato un esempio di configurazione per ciascun nodo della rete mesh.

Script di Configurazione per iptables:

```
# Flushing existing rules
sudo iptables -F
```

```
sudo iptables -t nat -F

# Allow traffic on loopback interface
sudo iptables -A INPUT -i lo -j ACCEPT

# Allow established and related incoming connections
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow OpenVPN traffic
sudo iptables -A INPUT -p udp --dport 1194 -j ACCEPT

# Allow WireGuard traffic
sudo iptables -A INPUT -p udp --dport 51820 -j ACCEPT

# Allow IPsec traffic
sudo iptables -A INPUT -p esp -j ACCEPT
sudo iptables -A INPUT -p ah -j ACCEPT
sudo iptables -A INPUT -p udp --dport 500 -j ACCEPT
sudo iptables -A INPUT -p udp --dport 4500 -j ACCEPT

# Allow traffic from VPN
sudo iptables -A INPUT -s 10.0.0.0/24 -j ACCEPT
sudo iptables -A FORWARD -s 10.0.0.0/24 -j ACCEPT
sudo iptables -A FORWARD -d 10.0.0.0/24 -j ACCEPT

# Drop everything else
sudo iptables -A INPUT -j DROP
```

Descrizione della Configurazione

1. **Reset delle Regole Esistenti**: Le regole correnti vengono eliminate per garantire una configurazione pulita.
2. **Loopback**: Il traffico sull'interfaccia di loopback (lo) viene sempre accettato per consentire comunicazioni locali.
3. **Traffico Stabilito**: Le connessioni già stabilite o correlate sono accettate per garantire la continuità delle comunicazioni.
4. **Traffico VPN**: Sono permesse le porte specifiche utilizzate dai protocolli VPN (OpenVPN, WireGuard e IPsec).
5. **VPN Interna**: Il traffico proveniente dalla rete mesh (10.0.0.0/24) è accettato sia in ingresso che per il forwarding.
6. **Politica di Default**: Tutto il traffico non esplicitamente consentito è bloccato.

Configurazione con nftables

nftables è il successore di iptables, progettato per fornire maggiore flessibilità e prestazioni migliori. Di seguito è fornito un esempio di configurazione equivalente utilizzando nftables.

Script di Configurazione per nftables:

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
    chain input {
```

```
        type filter hook input priority 0; policy drop;

        # Allow traffic on loopback interface
        iif "lo" accept

        # Allow established and related incoming connections
        ct state established,related accept

        # Allow OpenVPN traffic
        udp dport 1194 accept

        # Allow WireGuard traffic
        udp dport 51820 accept

        # Allow IPsec traffic
        ip protocol esp accept
        ip protocol ah accept
        udp dport 500 accept
        udp dport 4500 accept

        # Allow traffic from VPN
        ip saddr 10.0.0.0/24 accept
    }

    chain forward {
        type filter hook forward priority 0; policy drop;

        # Allow forwarding of VPN traffic
        ip saddr 10.0.0.0/24 accept
        ip daddr 10.0.0.0/24 accept
    }

    chain output {
        type filter hook output priority 0; policy accept;
    }
}
```

Descrizione della Configurazione:

1. **Cancellazione delle Regole Esistenti**: Il comando flush ruleset elimina tutte le regole precedenti.
2. **Tabella e Catene**: La configurazione è organizzata in catene di regole (input, forward, output) nella tabella inet filter.
3. **Regole per le VPN**: Sono accettate le porte specifiche di OpenVPN, WireGuard e IPsec, insieme al traffico proveniente dalla rete 10.0.0.0/24.
4. **Politiche di Default**: Il traffico non autorizzato è bloccato grazie alla politica di default drop.

Conclusioni

Questo rapporto tecnico ha descritto in modo dettagliato il processo di configurazione e implementazione di una rete mesh sicura composta da cinque nodi, connessi tramite tecnologie VPN e supportati da protocolli di routing dinamico. La rete è stata progettata per garantire sicurezza, resilienza e scalabilità in contesti complessi, dove la distribuzione dei nodi e la protezione delle comunicazioni rappresentano requisiti fondamentali.

La configurazione della rete è iniziata con OpenVPN, scelto per la sua robustezza e versatilità nel creare tunnel sicuri. Questa tecnologia ha fornito la base per stabilire connessioni protette tra i nodi, utilizzando chiavi e certificati per l'autenticazione e la crittografia dei dati. Successivamente, WireGuard è stato integrato come tecnologia VPN leggera ed efficiente, ottimizzando le prestazioni e semplificando la gestione della rete. A completare la configurazione VPN, è stato implementato Libreswan per supportare IPsec, garantendo compatibilità con standard consolidati e soddisfacendo requisiti di sicurezza aggiuntivi.

Per la gestione del routing dinamico, è stato utilizzato BIRD, configurato per supportare sia il protocollo BGP che OSPF. BGP è stato scelto per gestire le rotte tra i sistemi autonomi, assicurando che i dati venissero instradati attraverso i percorsi più appropriati tra i nodi. OSPF, invece, è stato utilizzato per il routing interno, ottimizzando i percorsi e garantendo una rapida convergenza in caso di variazioni nella topologia della rete. L'integrazione di questi protocolli ha permesso di creare una rete flessibile, capace di adattarsi dinamicamente ai cambiamenti e di mantenere una comunicazione stabile anche in presenza di guasti.

La sicurezza della rete è stata ulteriormente rafforzata attraverso la configurazione di firewall su ciascun nodo, utilizzando strumenti come iptables e nftables. Questi firewall sono stati configurati per implementare una politica di "default deny", consentendo solo il traffico autorizzato relativo alle VPN, al routing interno e ai servizi essenziali. La protezione è stata ottimizzata per prevenire accessi non autorizzati e garantire la sicurezza del traffico tra i nodi.

Il processo di implementazione è stato suddiviso in capitoli chiave, ciascuno dedicato a uno specifico aspetto della rete:

1. **Configurazione di OpenVPN**:

   Abbiamo avviato il progetto configurando OpenVPN per garantire una base solida e versatile per la connessione iniziale tra i nodi. Questa configurazione include l'utilizzo di chiavi e certificati per autenticare e crittografare il traffico.

2. **Configurazione di WireGuard**:

   Successivamente, abbiamo aggiunto WireGuard per sfruttare la sua efficienza e semplicità. La configurazione di WireGuard ha reso possibile l'ottimizzazione delle prestazioni, garantendo connessioni rapide e sicure tra i nodi.

3. **Configurazione di Libreswan (IPsec)**:

   Per supportare scenari che richiedono compatibilità con standard IPsec, abbiamo configurato Libreswan come ulteriore strumento VPN, integrandolo con le altre tecnologie.

4. **Routing Dinamico con BIRD**:

   Il routing dinamico è stato implementato utilizzando BIRD, configurato per supportare sia BGP che OSPF. Mentre BGP è stato utilizzato per gestire le rotte tra nodi autonomi, OSPF ha ottimizzato il routing interno, garantendo percorsi sempre aggiornati e una rapida convergenza.

5. **Configurazione del Firewall**:

   La protezione della rete è stata affrontata configurando firewall su ogni nodo, utilizzando sia iptables che nftables. Le regole definite assicurano che solo il traffico legittimo possa attraversare la rete, bloccando tutto ciò che non è esplicitamente autorizzato.

Una volta completata la configurazione, i servizi sono stati avviati per rendere operativa la rete.

Dopo aver configurato i file, è necessario avviare i servizi su ogni nodo per rendere operativa la rete. Di seguito sono riportati i comandi essenziali per avviare i servizi VPN, di routing e di sicurezza:

1. **OpenVPN**:
   - Su Nodo 1 (N1):

     ```
     sudo systemctl start openvpn@server
     ```

   - Su Nodi 2, 3, 4, 5:

     ```
     sudo systemctl start openvpn@client
     ```

2. **WireGuard**:
   - Su tutti i nodi:

     ```
     sudo wg-quick up wg0
     ```

3. **Libreswan (IPsec)**:
   - Su tutti i nodi:

     ```
     sudo ipsec start
     ```

4. **BIRD**:
   - Su tutti i nodi:

     ```
     sudo systemctl start bird
     ```

OpenVPN è stato avviato con configurazioni distinte per server e client, mentre WireGuard è stato attivato tramite il comando `wg-quick`. Libreswan, configurato per IPsec, è stato avviato su ciascun nodo, e BIRD è stato reso operativo per garantire il funzionamento dei protocolli BGP e OSPF. Questi passi hanno consentito di verificare la piena operatività della rete, assicurandone il corretto funzionamento e il rispetto dei requisiti di sicurezza e affidabilità.

In sintesi, questa configurazione ha realizzato una rete mesh sicura, scalabile e flessibile, basata su tecnologie moderne e consolidate. OpenVPN, WireGuard e Libreswan hanno garantito un livello elevato di protezione per le comunicazioni, mentre il routing dinamico implementato con BIRD ha assicurato una gestione efficiente del traffico. La configurazione del firewall ha completato l'architettura, proteggendo i nodi e il traffico interno. Questa rete rappresenta una soluzione robusta per applicazioni in ambienti distribuiti, infrastrutture critiche e sistemi che richiedono una comunicazione sicura e stabile. Grazie alla modularità e alla scalabilità della configurazione, la rete può essere facilmente adattata a scenari specifici e ampliata con l'aggiunta di nuovi nodi. Questo rapporto offre una guida pratica e completa per implementare reti mesh sicure e moderne, rispondendo alle esigenze di infrastrutture distribuite complesse.

Bibliografia

1. GitHub. **OpenVPN**. Disponibile online: https://github.com/OpenVPN, ultima consultazione il 11 Gennaio 2025.
2. WireGuard Official Site. **WireGuard**. Disponibile online: https://www.wireguard.com/, ultima consultazione il 11 Gennaio 2025.
3. RFC 4271. **A Border Gateway Protocol 4 (BGP-4)**. Disponibile online: https://www.rfc-editor.org/rfc/rfc4271, ultima consultazione il 11 Gennaio 2025.
4. Cisco. **Open Shortest Path First (OSPF)**. Disponibile online: https://www.cisco.com/c/en/us/products/ios-nx-os-software/open-shortest-path-first-ospf/index.html, ultima consultazione il 11 Gennaio 2025.
5. Cisco. **IP Routing: OSPF Configuration Guide**. Disponibile online: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html, ultima consultazione il 11 Gennaio 2025.
6. HTML.it. **OpenVPN, guida al server VPN open source**. Disponibile online: https://www.html.it/articoli/openvpn-un-server-vpn-open-source/, ultima consultazione il 11 Gennaio 2025.
7. openvpn.net. **Creating configuration files for server and clients**. Disponibile online: https://openvpn.net/community-resources/creating-configuration-files-for-server-and-clients/, ultima consultazione il 11 Gennaio 2025.
8. ITIGIC. Tutorial OpenVPN: installazione, configurazione del server VPN e connessione. Disponibile online: https://itigic.com/it/openvpn-tutorial-installation-vpn-server-configuration/, ultima consultazione il 11 Gennaio 2025.
9. IBM. **Configure IPSec VPN between AIX and LINUX using Libreswan : A Step-by-Step Guide**. Disponibile online: https://community.ibm.com/community/user/power/blogs/rajya-lakshmi-marathu/2024/11/17/create-ipsec-vpn-between-aix-linux-with-libreswan, ultima consultazione il 11 Gennaio 2025.
10. Oracle. **Accesso ad altri cloud con Libreswan**. Disponibile online: https://docs.oracle.com/it-it/iaas/Content/Network/Concepts/libreswan.htm, ultima consultazione il 11 Gennaio 2025.
11. Oracle Linux. **How to Configure IPsec to Secure Site-to-Site Communications Using Libreswan.** Disponibile online: https://support.oracle.com/knowledge/Oracle%20Linux%20and%20Virtualization/2372172_1.html, ultima consultazione il 11 Gennaio 2025.
12. Libreswan Official Site. **Host to host VPN**. Disponibile online: https://libreswan.org/wiki/Host_to_host_VPN, ultima consultazione il 11 Gennaio 2025.
13. BIRD Official Site. **The BIRD Internet Routing Daemon**. Disponibile online: https://bird.network.cz/, ultima consultazione il 11 Gennaio 2025.