Router-on-a-stick virtualizzato: firewall guest su hypervisor Debian (e hardening OPSEC)

Router-on-a-stick virtualizzato: firewall guest su hypervisor Debian (e hardening OPSEC)

Nelle architetture di rete casalinghe o nei lab di sicurezza, l'approccio standard del sistemista improvvisato è installare il firewall direttamente sull'hardware host o, peggio ancora, esporre l'interfaccia di management dell'hypervisor su qualsiasi subnet disponibile. Niente di più sbagliato. Se un servizio esposto in DMZ o un client compromesso sulla LAN fidata riesce a fare un exploit, ha accesso diretto alle chiavi dell'hypervisor (KVM, Proxmox o quel che sia). Game over.

La soluzione elegante e sicura è segregare i ruoli: un hypervisor Linux Debian minimale che si occupa solo di gestire i bridge virtuali (senza metterci la faccia, ovvero senza assegnarsi IP) e una macchina virtuale guest KVM dedicata che fa da vero e proprio firewall/router centralizzato.

Oggi vediamo come configurare in modo pulito e anonimo un'architettura **Router-on-a-stick virtualizzata** sotto Debian, con un occhio di riguardo all'hardening OPSEC.

Lo Scenario: Router-on-a-stick

L'host fisico dispone di una sola interfaccia di rete Ethernet (`eth0`). Questa interfaccia è collegata a una porta trunk di uno switch managed, su cui transitano diverse VLAN taggate:

- **VLAN 10**: Rete Esterna (WAN)
- **VLAN 50**: Zona Demilitarizzata (DMZ)
- **VLAN 100**: Rete Fidata (Trust LAN)
- **VLAN 150**: Rete anonimizzata (Tor LAN)

Il nostro scopo è far confluire tutte le VLAN dentro la macchina virtuale firewall, la quale si occuperà di fare routing, NATting, filtraggio dei pacchetti (`nftables`) e redirect trasparenti su Tor.

Preparazione dell'Host (Pacchetti da installare)

Per prima cosa dobbiamo trasformare la nostra installazione Debian minimale in un hypervisor KVM e abilitare gli strumenti di bridging di rete.

Installiamo i pacchetti necessari tramite `apt`:

```
sudo apt install qemu-system qemu-utils libvirt-daemon-system libvirt-clients bridge-utils
```

(Nota per architetture ARM64 come Raspberry Pi: installate `qemu-system-aarch64` al posto del generico `qemu-system` per sfruttare la virtualizzazione nativa).

Assicuriamoci che i servizi di virtualizzazione siano attivi e abilitati al boot:

```
sudo systemctl enable --now libvirtd
```

---

1. Configurazione dell'Host (Hypervisor Debian)

Sull'host Debian configuriamo le sotto-interfacce VLAN e le associamo a rispettivi bridge virtuali. La regola d'oro per l'hardening è: **l'host non deve avere un IP sulle reti che non deve gestire direttamente.**

Modifichiamo `/etc/network/interfaces` sull'host:

```
source /etc/network/interfaces.d/*

# Configurazione delle sotto-interfacce VLAN (senza IP logico)
iface eth0 inet manual

iface eth0.10 inet manual
iface eth0.50 inet manual
```

```
iface eth0.100 inet manual
iface eth0.150 inet manual

# Abilitiamo i bridge all'avvio
auto br0
auto br1
auto br2
auto br3

# BRIDGE WAN (VLAN 10) - L'host NON ha IP qui
iface br0 inet manual
        bridge_ports eth0.10
        bridge_stp off

# BRIDGE DMZ (VLAN 50) - L'host NON ha IP qui
iface br1 inet manual
        bridge_ports eth0.50
        bridge_stp off

# BRIDGE TRUST (VLAN 100) - L'host ha l'IP di management solo qui
iface br2 inet static
        address 10.152.0.254
        netmask 255.255.255.0
        broadcast 10.152.0.255
        bridge_ports eth0.100
        bridge_stp off
        bridge_waitport 0

# BRIDGE TOR (VLAN 150) - L'host NON ha IP qui
iface br3 inet manual
        bridge_ports eth0.150
        bridge_stp off
        bridge_waitport 0
```

Perché impostare "manual" sui bridge sensibili?

Notate come `br0` (WAN), `br1` (DMZ) e `br3` (Tor) siano configurati in modalità `manual` (senza indirizzo IP logico per l'host). Questo è cruciale per la sicurezza:

- I pacchetti Ethernet delle VM transitano comunque attraverso il bridge (che agisce come uno switch Layer 2 virtuale puro).
- L'host hypervisor non ha uno stack IP attivo su quelle subnet. Se un attaccante compromette un web server in DMZ (`br1`), non troverà alcun indirizzo IP dell'hypervisor da scansionare o attaccare su quel segmento. L'hypervisor è invisibile.

- L'host è raggiungibile solo sulla rete Trust (`br2` all'IP `10.152.0.254`), isolando la console di gestione SSH dal resto del mondo.

---

2. Configurazione del Guest (KVM Firewall VM)

Nel gestore di virtualizzazione (es. Libvirt / QEMU-KVM), creiamo la macchina virtuale del firewall e associamo ad essa le interfacce virtuali collegate direttamente ai bridge dell'host:

- Interfaccia `virtio` 1 → collegata al bridge `br0` (WAN)
- Interfaccia `virtio` 2 → collegata al bridge `br1` (DMZ)
- Interfaccia `virtio` 3 → collegata al bridge `br2` (Trust)
- Interfaccia `virtio` 4 → collegata al bridge `br3` (Tor)

All'interno della VM firewall (anch'essa idealmente una Debian minimale), le interfacce verranno rilevate come `enp1s0`, `enp8s0`, `enp9s0`, `enp12s0` (i nomi possono variare in base alla configurazione PCI).

Configuriamo `/etc/network/interfaces` nella VM firewall:

```
auto lo
iface lo inet loopback

# Interfaccia Esterna (WAN)
auto enp1s0
iface enp1s0 inet static
        address 10.255.0.5
        netmask 255.255.255.0
        broadcast 10.255.0.255
        gateway 10.255.0.1
        dns-nameservers 1.1.1.1

# Interfaccia DMZ (Gateway per la subnet DMZ)
auto enp8s0
iface enp8s0 inet static
        address 10.153.0.1
        netmask 255.255.255.0
        broadcast 10.153.0.255

# Interfaccia Trust (Gateway per la subnet locale fidata)
auto enp9s0
iface enp9s0 inet static
```

```
        address 10.152.0.1
        netmask 255.255.255.0
        broadcast 10.152.0.255

# Interfaccia TOR (Gateway per i client instradati sotto Tor)
auto enp12s0
iface enp12s0 inet static
        address 10.254.0.1
        netmask 255.255.255.0
        broadcast 10.254.0.255
```

Conclusioni operative

Con questo schema, la VM firewall diventa l'unico default gateway delle macchine collegate fisicamente alle VLAN 50, 100 e 150. Qualsiasi pacchetto diretto a internet dovrà passare sotto le forche caudine del firewall virtuale, che deciderà se fare NAT, inoltrare il traffico in chiaro o forzarlo in modalità trasparente dentro la rete Tor.

L'hypervisor Debian rimane silente e al sicuro da scansioni esterne o compromissioni locali. OPSEC solida, fatica minima.

[Che il talpone sia con voi]

Fluxer Icon

Canale di discussione e supporto

Canale di discussione e supporto. Per confrontarti su configurazioni o altro entra su fluxer.dreadful.work (https://fluxer.dreadful.work/invite/dreadful).