NFTABLES

¿QUÉ ES?

Nftables es un framework que sustituye al antiguo iptables, está implementado desde las versiones de kernel linux 3.13.

Utilidad “nft” con una sintaxis diferente de iptables.

Usa la infraestructura de Netfilter.

Tiene compatibilidad con las instrucciones de iptables.

Aun está bajo desarrollo.

DIFERENCIAS CON IPTABLES

- La sintaxis, siendo esta más limpia, inspirada en tcpdump.
- Tablas y cadenas totalmente configurables, pudiendo ser propias.
- Administración simplificada para conjuntos IPv4/IPv6
- Se pueden especificar varias acciones en una sola línea .
- Contadores opcionales, al contrario que en iptables.
- Infraestructura de conjuntos genéricos (rulesets).
- No es necesario el uso de guiones para el uso de flags.

TABLAS

Por el momento hay 6 tipos de tablas, dependiendo de su familia:
(Ip, Arp, Ip6, Bridge, Inet, netdev)

La tabla Inet es una tabla híbrida de IPv4 + IPv6, esto significa que las reglas de esta tabla servirán para filtrar el tráfico ipv4 e ipv6.

Las sintaxis para añadir una tabla es la siguiente:

%nft add table [familia] <nombre>

Para ver el listado de tablas:

%nft list tables

HOOKS

Los posibles tipos de hooks que podemos utilizar son:

- **Prerouting**: Filtra los paquetes antes de tomar la decisión de enrutamiento.

- **Input:** Filtra los paquetes que vienen a nuestro sistema.

- **Forward:** Filtra los paquetes que pasan por la máquina sin ser su destino.

- **Output:** Paquetes que se originan en la máquina y salen con destino otra.

- **Postrouting:** Filtra después de la decisión de enrutamiento.

- **Ingress(familia netdev) :** Desde el kernel 4.2 , se puede filtrar el tráfico antes de prerouting, después de que el paquete se mande desde el controlados NIC.

CADENAS

Es necesario crear las cadenas base ya que no vienen predefinidas, indicando el tipo de hook que utilizará la cadena para filtrar tráfico.

> *%nft add chain [familia] <nombre-tabla> <nombre-cadena> {type <tipo> hook <hook> priority <valor> \; policy <política> }*

Hay 3 tipos de cadenas:

- Nat → Se utiliza para hacer traducciones de direcciones de red (NAT).
- Filter → Se utiliza para filtrar paquetes.
- Route → Se uso es reencaminar paquetes (Similar de mangle en iptables)

REGLAS

La creación de reglas tiene una sintaxis sencilla, por ejemplo:
Filtrando paquetes según su cabecera IPv4:

%nft add rule filter input ip saddr 192.168.1.100 ip daddr 192.168.1.1 counter accept

Filtrando según el puerto TCP/UDP:

% nft add rule filter input tcp dport 1-1024 counter drop

Según interfaz de entrada y de salida:

%nft add rule filter input iifname eth0 tcp dport 80 counter accept

JUMP

Esta utilidad sirve para saltar a otras cadenas que hayamos creado.

Ejemplo:

```
table ip filter {
        chain other-chain {
                tcp dport 80 counter drop
                tcp dport 22 counter drop
            }
        chain input { type filter hook input priority 0; policy accept;
                ip saddr 1.1.1.1 ip daddr 2.2.2.2 jump other-chain
            }
}
```