Red Hat Enterprise Linux 8

Asegurar las redes

Configuración de redes seguras y comunicación en red

Last Updated: 2021-03-05

Red Hat Enterprise Linux 8 Asegurar las redes

Configuración de redes seguras y comunicación en red

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

Legal Notice

Resumen

Este título ayuda a los administradores a proteger las redes, las máquinas conectadas y la comunicación de red contra diversos ataques.

Table of Contents

HACER QUE EL CÓDIGO ABIERTO SEA MÁS INCLUSIVO

Red Hat se compromete a sustituir el lenguaje problemático en nuestro código, documentación y propiedades web. Estamos empezando con estos cuatro términos: maestro, esclavo, lista negra y lista blanca. Debido a la enormidad de este esfuerzo, estos cambios se implementarán gradualmente a lo largo de varias versiones próximas. Para más detalles, consulte el mensaje de nuestro CTO Chris Wright .

PROPORCIONAR COMENTARIOS SOBRE LA DOCUMENTACIÓN DE RED HAT

Agradecemos su opinión sobre nuestra documentación. Por favor, díganos cómo podemos mejorarla. Para ello:

- Para comentarios sencillos sobre pasajes concretos:
  1. Asegúrese de que está viendo la documentación en el formato *Multi-page HTML*. Además, asegúrese de ver el botón **Feedback** en la esquina superior derecha del documento.
  2. Utilice el cursor del ratón para resaltar la parte del texto que desea comentar.
  3. Haga clic en la ventana emergente **Add Feedback** que aparece debajo del texto resaltado.
  4. Siga las instrucciones mostradas.
- Para enviar comentarios más complejos, cree un ticket de Bugzilla:
  1. Vaya al sitio web de Bugzilla.
  2. Como componente, utilice **Documentation**.
  3. Rellene el campo **Description** con su sugerencia de mejora. Incluya un enlace a la(s) parte(s) pertinente(s) de la documentación.
  4. Haga clic en **Submit Bug**.

CAPÍTULO 1. USO DE COMUNICACIONES SEGURAS ENTRE DOS SISTEMAS CON OPENSSH

SSH (Secure Shell) es un protocolo que proporciona comunicaciones seguras entre dos sistemas utilizando una arquitectura cliente-servidor y permite a los usuarios iniciar la sesión en los sistemas anfitriones del servidor de forma remota. A diferencia de otros protocolos de comunicación remota, como FTP o Telnet, SSH cifra la sesión de inicio de sesión, lo que impide que los intrusos recojan las contraseñas no cifradas de la conexión.

Red Hat Enterprise Linux incluye los paquetes básicos **OpenSSH**: el paquete general **openssh**, el paquete **openssh-server** y el paquete **openssh-clients**. Tenga en cuenta que los paquetes **OpenSSH** requieren el paquete **OpenSSL openssl-libs** , que instala varias bibliotecas criptográficas importantes que permiten a **OpenSSH** proporcionar comunicaciones cifradas.

1.1. SSH Y OPENSSH

SSH (Secure Shell) es un programa para entrar en una máquina remota y ejecutar comandos en esa máquina. El protocolo SSH proporciona comunicaciones seguras y encriptadas entre dos hosts no confiables a través de una red insegura. También puede reenviar conexiones X11 y puertos TCP/IP arbitrarios a través del canal seguro.

El protocolo SSH mitiga las amenazas de seguridad, como la interceptación de la comunicación entre dos sistemas y la suplantación de un determinado host, cuando se utiliza para el inicio de sesión de shell remoto o la copia de archivos. Esto se debe a que el cliente y el servidor SSH utilizan firmas digitales para verificar sus identidades. Además, toda la comunicación entre los sistemas cliente y servidor está cifrada.

OpenSSH es una implementación del protocolo SSH soportada por varios sistemas operativos Linux, UNIX y similares. Incluye los archivos centrales necesarios para el cliente y el servidor de OpenSSH. La suite OpenSSH consiste en las siguientes herramientas de espacio de usuario:

- **ssh** es un programa de acceso remoto (cliente SSH)
- **sshd** es un demonio SSH **OpenSSH**
- **scp** es un programa de copia remota segura de archivos
- **sftp** es un programa de transferencia segura de archivos
- **ssh-agent** es un agente de autenticación para el almacenamiento de claves privadas
- **ssh-add** añade identidades de clave privada a **ssh-agent**
- **ssh-keygen** genera, gestiona y convierte las claves de autenticación para **ssh**
- **ssh-copy-id** es un script que añade claves públicas locales al archivo **authorized_keys** en un servidor SSH remoto
- **ssh-keyscan** - recoge las claves públicas de host SSH

Actualmente existen dos versiones de SSH: la versión 1 y la versión 2, más reciente. La suite **OpenSSH** en Red Hat Enterprise Linux 8 sólo soporta la versión 2 de SSH, que tiene un algoritmo de intercambio de claves mejorado que no es vulnerable a los exploits conocidos de la versión 1.

OpenSSH, como uno de los subsistemas criptográficos centrales de RHEL, utiliza políticas criptográficas para todo el sistema. Esto asegura que los conjuntos de cifrado y los algoritmos

criptográficos débiles están desactivados en la configuración por defecto. Para ajustar la política, el administrador debe utilizar el comando **update-crypto-policies** para hacer la configuración más estricta o más floja o excluir manualmente las políticas criptográficas de todo el sistema.

El conjunto **OpenSSH** utiliza dos conjuntos diferentes de archivos de configuración: los de los programas cliente (es decir, **ssh**, **scp**, y **sftp**), y los del servidor (el demonio **sshd**). La información de configuración de SSH para todo el sistema se almacena en el directorio **/etc/ssh/**. La información de configuración SSH específica del usuario se almacena en **~/.ssh/** en el directorio de inicio del usuario. Para una lista detallada de los archivos de configuración de OpenSSH, vea la sección **FILES** en la página man **sshd(8)**.

Recursos adicionales

- Páginas de manual para el tema **ssh** listadas por el comando **man -k ssh**.
- Uso de políticas criptográficas en todo el sistema .

1.2. CONFIGURAR E INICIAR UN SERVIDOR OPENSSH

Utilice el siguiente procedimiento para una configuración básica que puede ser necesaria para su entorno y para iniciar un servidor **OpenSSH**. Tenga en cuenta que después de la instalación por defecto de RHEL, el demonio **sshd** ya está iniciado y las claves del servidor se crean automáticamente.

Requisitos previos

- El paquete **openssh-server** está instalado.

Procedimiento

1. Inicie el demonio **sshd** en la sesión actual y configúrelo para que se inicie automáticamente al arrancar:

```
# systemctl start sshd
# systemctl enable sshd
```

2. Para especificar direcciones diferentes a las predeterminadas **0.0.0.0** (IPv4) o **::** (IPv6) para la directiva **ListenAddress** en el archivo de configuración **/etc/ssh/sshd_config** y utilizar una configuración de red dinámica más lenta, añada la dependencia de la unidad de destino **network-online.target** al archivo de unidad **sshd.service**. Para ello, cree el archivo **/etc/systemd/system/sshd.service.d/local.conf** con el siguiente contenido:

```
[Unit]
Wants=network-online.target
After=network-online.target
```

3. Revise si la configuración del servidor **OpenSSH** en el archivo de configuración **/etc/ssh/sshd_config** cumple con los requisitos de su escenario.

4. Opcionalmente, cambie el mensaje de bienvenida que su servidor **OpenSSH** muestra antes de que un cliente se autentique editando el archivo **/etc/issue**, por ejemplo:

```
Welcome to ssh-server.example.com
Warning: By accessing this server, you agree to the referenced terms and conditions.
```

Asegúrese de que la opción **Banner** no está comentada en **/etc/ssh/sshd_config** y su valor contiene **/etc/issue**:

```
# less /etc/ssh/sshd_config | grep Banner
Banner /etc/issue
```

Tenga en cuenta que para cambiar el mensaje que se muestra después de un inicio de sesión exitoso tiene que editar el archivo **/etc/motd** en el servidor. Consulte la página man **pam_motd** para obtener más información.

5. Vuelva a cargar la configuración de **systemd** y reinicie **sshd** para aplicar los cambios:

```
# systemctl daemon-reload
# systemctl restart sshd
```

Pasos de verificación

1. Compruebe que el demonio **sshd** se está ejecutando:

```
# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-11-18 14:59:58 CET; 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1149 (sshd)
    Tasks: 1 (limit: 11491)
   Memory: 1.9M
   CGroup: /system.slice/sshd.service
           └─1149 /usr/sbin/sshd -D -oCiphers=aes128-ctr,aes256-ctr,aes128-cbc,aes256-cbc -
oMACs=hmac-sha2-256,>

Nov 18 14:59:58 ssh-server-example.com systemd[1]: Starting OpenSSH server daemon...
Nov 18 14:59:58 ssh-server-example.com sshd[1149]: Server listening on 0.0.0.0 port 22.
Nov 18 14:59:58 ssh-server-example.com sshd[1149]: Server listening on :: port 22.
Nov 18 14:59:58 ssh-server-example.com systemd[1]: Started OpenSSH server daemon.
```

2. Conéctese al servidor SSH con un cliente SSH.

```
# ssh user@ssh-server-example.com
ECDSA key fingerprint is SHA256:dXbaS0RG/UzlTTku8GtXSz0S1++lPegSy31v3L/FAEc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ssh-server-example.com' (ECDSA) to the list of known hosts.

user@ssh-server-example.com's password:
```

Recursos adicionales

- **sshd(8)** y **sshd_config(5)** páginas man

1.3. USO DE PARES DE CLAVES EN LUGAR DE CONTRASEÑAS PARA LA AUTENTICACIÓN SSH

Para mejorar aún más la seguridad del sistema, genere pares de claves SSH y luego aplique la autenticación basada en claves deshabilitando la autenticación por contraseña.

1.3.1. Configuración de un servidor OpenSSH para la autenticación basada en claves

Siga estos pasos para configurar su servidor OpenSSH para aplicar la autenticación basada en claves.

Requisitos previos

- El paquete **openssh-server** está instalado.
- El demonio **sshd** se está ejecutando en el servidor.

Procedimiento

1. Abra la configuración de **/etc/ssh/sshd_config** en un editor de texto, por ejemplo:

```
# vi /etc/ssh/sshd_config
```

2. Cambie la opción **PasswordAuthentication** por **no**:

```
PasswordAuthentication no
```

   En un sistema que no sea una instalación nueva por defecto, compruebe que no se ha configurado **PubkeyAuthentication no** y que la directiva **ChallengeResponseAuthentication** está establecida en **no**. Si está conectado de forma remota, sin utilizar la consola o el acceso fuera de banda, pruebe el proceso de inicio de sesión basado en la clave antes de desactivar la autenticación por contraseña.

3. Para utilizar la autenticación basada en claves con los directorios personales montados en NFS, active el booleano **use_nfs_home_dirs** SELinux:

```
# setsebool -P use_nfs_home_dirs 1
```

4. Vuelva a cargar el demonio **sshd** para aplicar los cambios:

```
# systemctl reload sshd
```

Recursos adicionales

- **sshd(8)**, **sshd_config(5)**, y **setsebool(8)** páginas de manual

1.3.2. Generación de pares de claves SSH

Utilice este procedimiento para generar un par de claves SSH en un sistema local y para copiar la clave pública generada en un servidor **OpenSSH**. Si el servidor está configurado como corresponde, podrá iniciar sesión en el servidor **OpenSSH** sin necesidad de proporcionar ninguna contraseña.

IMPORTANTE

Si completa los siguientes pasos como **root**, sólo **root** podrá utilizar las llaves.

Procedimiento

1. Para generar un par de claves ECDSA para la versión 2 del protocolo SSH:

```
$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/joesec/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/joesec/.ssh/id_ecdsa.
Your public key has been saved in /home/joesec/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:Q/x+qms4j7PCQ0qFd09iZEFHA+SqwBKRNaU72oZfaCI
joesec@localhost.example.com
The key's randomart image is:
+---[ECDSA 256]---+
|.oo..o=++        |
|.. o .oo .       |
|. .. o. o        |
|....o.+...       |
|o.oo.o +S .      |
|.=.+.   .o       |
|E.*+.  .  . .    |
|.=..+ +..  o     |
|  .  oo*+o.      |
+----[SHA256]-----+
```

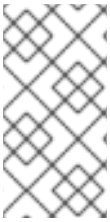
También puede generar un par de claves RSA utilizando la opción **-t rsa** con el comando **ssh-keygen** o un par de claves Ed25519 introduciendo el comando **ssh-keygen -t ed25519**.

2. Para copiar la clave pública en una máquina remota:

```
$ ssh-copy-id joesec@ssh-server-example.com
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are
already installed
joesec@ssh-server-example.com's password:
...
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'joesec@ssh-server-example.com'" and check to
make sure that only the key(s) you wanted were added.
```

Si no utiliza el programa **ssh-agent** en su sesión, el comando anterior copia la clave pública más recientemente modificada **~/.ssh/id*.pub** si aún no está instalada. Para especificar otro archivo de clave pública o para dar prioridad a las claves en archivos sobre las claves almacenadas en la memoria por **ssh-agent**, utilice el comando **ssh-copy-id** con la opción **-i**.



NOTA

Si reinstalas tu sistema y quieres conservar los pares de claves generados anteriormente, haz una copia de seguridad del directorio **~/.ssh/**. Después de reinstalar, cópialo de nuevo en tu directorio principal. Puedes hacer esto para todos los usuarios de tu sistema, incluyendo **root**.

Pasos de verificación

1. Inicie sesión en el servidor OpenSSH sin proporcionar ninguna contraseña:

```
$ ssh joesec@ssh-server-example.com
Welcome message.
...
Last login: Mon Nov 18 18:28:42 2019 from ::1
```

Recursos adicionales

- **ssh-keygen(1)** y **ssh-copy-id(1)** páginas man

1.4. USO DE CLAVES SSH ALMACENADAS EN UNA TARJETA INTELIGENTE

Red Hat Enterprise Linux 8 le permite utilizar claves RSA y ECDSA almacenadas en una tarjeta inteligente en clientes OpenSSH. Utilice este procedimiento para habilitar la autenticación utilizando una tarjeta inteligente en lugar de utilizar una contraseña.

Requisitos previos

- En el lado del cliente, el paquete **opensc** está instalado y el servicio **pcscd** está funcionando.

Procedimiento

1. Enumerar todas las claves proporcionadas por el módulo PKCS #11 de OpenSC incluyendo sus URIs PKCS #11 y guardar el resultado en el archivo *keys.pub*:

```
$ ssh-keygen -D pkcs11: > keys.pub
$ ssh-keygen -D pkcs11:
ssh-rsa AAAAB3NzaC1yc2E...KKZMzcQZzx
pkcs11:id=%02;object=SIGN%20pubkey;token=SSH%20key;manufacturer=piv_II?module-
path=/usr/lib64/pkcs11/opensc-pkcs11.so
ecdsa-sha2-nistp256 AAA...J0hkYnnsM=
pkcs11:id=%01;object=PIV%20AUTH%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
```

2. Para habilitar la autenticación mediante una tarjeta inteligente en un servidor remoto (*example.com*), transfiera la clave pública al servidor remoto. Utilice el comando **ssh-copy-id** con *keys.pub* creado en el paso anterior:

```
$ ssh-copy-id -f -i keys.pub username@example.com
```

3. Para conectarse a *example.com* utilizando la clave ECDSA de la salida del comando **ssh-keygen -D** en el paso 1, puede utilizar sólo un subconjunto de la URI, que hace referencia a su clave de forma exclusiva, por ejemplo:

```
$ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" example.com
Enter PIN for 'SSH key':
[example.com] $
```

4. Puede utilizar la misma cadena URI en el archivo **~/.ssh/config** para que la configuración sea permanente:

```
$ cat ~/.ssh/config
```

```
IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
$ ssh example.com
Enter PIN for 'SSH key':
[example.com] $
```

Dado que OpenSSH utiliza el wrapper **p11-kit-proxy** y el módulo PKCS #11 de OpenSC está registrado en PKCS#11 Kit, puede simplificar los comandos anteriores:

```
$ ssh -i "pkcs11:id=%01" example.com
Enter PIN for 'SSH key':
[example.com] $
```

Si se omite la parte **id=** de un URI PKCS #11, OpenSSH carga todas las claves que están disponibles en el módulo proxy. Esto puede reducir la cantidad de escritura requerida:

```
$ ssh -i pkcs11: example.com
Enter PIN for 'SSH key':
[example.com] $
```

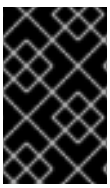
Recursos adicionales

- Fedora 28: Mejor soporte para tarjetas inteligentes en OpenSSH
- **p11-kit(8)** página de manual
- **ssh(1)** página de manual
- **ssh-keygen(1)** página de manual
- **opensc.conf(5)** página de manual
- **pcscd(8)** página de manual

1.5. CÓMO HACER QUE OPENSSH SEA MÁS SEGURO

Los siguientes consejos le ayudarán a aumentar la seguridad cuando utilice OpenSSH. Tenga en cuenta que los cambios en el archivo de configuración de **/etc/ssh/sshd_config** OpenSSH requieren la recarga del demonio **sshd** para que surtan efecto:

```
# systemctl reload sshd
```



IMPORTANTE

La mayoría de los cambios en la configuración del refuerzo de la seguridad reducen la compatibilidad con los clientes que no admiten algoritmos o conjuntos de cifrado actualizados.

Desactivación de los protocolos de conexión inseguros

- Para que SSH sea realmente eficaz, hay que evitar el uso de protocolos de conexión inseguros que sean sustituidos por el conjunto **OpenSSH**. De lo contrario, la contraseña de un usuario podría estar protegida usando SSH para una sesión sólo para ser capturada más tarde cuando

se conecte usando Telnet. Por esta razón, considere deshabilitar los protocolos inseguros, como telnet, rsh, rlogin y ftp.

Activación de la autenticación basada en clave y desactivación de la autenticación basada en contraseña

- Desactivar las contraseñas para la autenticación y permitir sólo los pares de claves reduce la superficie de ataque y también podría ahorrar tiempo a los usuarios. En los clientes, genere pares de claves utilizando la herramienta **ssh-keygen** y utilice la utilidad **ssh-copy-id** para copiar las claves públicas de los clientes en el servidor **OpenSSH**. Para desactivar la autenticación basada en contraseña en su servidor OpenSSH, edite **/etc/ssh/sshd_config** y cambie la opción **PasswordAuthentication** por **no**:

```
PasswordAuthentication no
```

Tipos de claves

- Aunque el comando **ssh-keygen** genera un par de claves RSA por defecto, puedes indicarle que genere claves ECDSA o Ed25519 utilizando la opción **-t**. El ECDSA (Algoritmo de Firma Digital de Curva Elíptica) ofrece un mejor rendimiento que el RSA con una fuerza de clave simétrica equivalente. También genera claves más cortas. El algoritmo de clave pública Ed25519 es una implementación de curvas de Edwards retorcidas que es más segura y también más rápida que RSA, DSA y ECDSA.
  OpenSSH crea automáticamente las claves de host del servidor RSA, ECDSA y Ed25519 si no las tiene. Para configurar la creación de claves de host en RHEL 8, utilice el servicio instanciado **sshd-keygen@.service**. Por ejemplo, para desactivar la creación automática del tipo de clave RSA:

```
# systemctl mask sshd-keygen@rsa.service
```

- Para excluir determinados tipos de claves para las conexiones SSH, comente las líneas correspondientes en **/etc/ssh/sshd_config** y vuelva a cargar el servicio **sshd**. Por ejemplo, para permitir sólo las claves de host Ed25519:

```
# HostKey /etc/ssh/ssh_host_rsa_key
# HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
```

Puerto no predeterminado

- Por defecto, el demonio **sshd** escucha en el puerto TCP 22. Cambiar el puerto reduce la exposición del sistema a ataques basados en el escaneo automático de la red y, por tanto, aumenta la seguridad a través de la oscuridad. Puede especificar el puerto utilizando la directiva **Port** en el archivo de configuración **/etc/ssh/sshd_config**.
  También tienes que actualizar la política por defecto de SELinux para permitir el uso de un puerto no predeterminado. Para ello, utilice la herramienta **semanage** del paquete **policycoreutils-python-utils**:

```
# semanage port -a -t ssh_port_t -p tcp port_number
```

  Además, actualice la configuración de **firewalld**:

```
# firewall-cmd --add-port port_number/tcp
# firewall-cmd --runtime-to-permanent
```

En los comandos anteriores, sustituya *port_number* por el nuevo número de puerto especificado mediante la directiva **Port**.

No hay acceso a la raíz

- Si su caso de uso particular no requiere la posibilidad de iniciar sesión como usuario root, debería considerar establecer la directiva de configuración **PermitRootLogin** a **no** en el archivo **/etc/ssh/sshd_config**. Al deshabilitar la posibilidad de iniciar sesión como usuario root, el administrador puede auditar qué usuarios ejecutan qué comandos privilegiados después de iniciar sesión como usuarios normales y luego obtener derechos de root.
  Como alternativa, configure **PermitRootLogin** en **prohibit-password**:

  ```
  PermitRootLogin prohibir-contraseña
  ```

  Esto refuerza el uso de la autenticación basada en claves en lugar del uso de contraseñas para iniciar la sesión como root y reduce los riesgos al evitar los ataques de fuerza bruta.

Uso de la extensión X Security

- El servidor X en los clientes de Red Hat Enterprise Linux no proporciona la extensión X Security. Por lo tanto, los clientes no pueden solicitar otra capa de seguridad cuando se conectan a servidores SSH no confiables con el reenvío X11. La mayoría de las aplicaciones no pueden ejecutarse con esta extensión habilitada de todos modos.
  Por defecto, la opción **ForwardX11Trusted** en el archivo **/etc/ssh/ssh_config.d/05-redhat.conf** se establece en **yes**, y no hay diferencia entre el comando **ssh -X remote_machine** (host no confiable) y **ssh -Y remote_machine** (host confiable).

  Si su escenario no requiere la función de reenvío de X11 en absoluto, establezca la directiva **X11Forwarding** en el archivo de configuración **/etc/ssh/sshd_config** a **no**.

Restringir el acceso a usuarios, grupos o dominios específicos

- Las directivas **AllowUsers** y **AllowGroups** en el archivo de configuración del servidor **/etc/ssh/sshd_config** le permiten permitir sólo a ciertos usuarios, dominios o grupos conectarse a su servidor OpenSSH. Puede combinar **AllowUsers** y **AllowGroups** para restringir el acceso con mayor precisión, por ejemplo:

  ```
  AllowUsers *@192.168.1.*,*@10.0.0.*,!*@192.168.1.2
  AllowGroups example-group
  ```

  Las líneas de configuración anteriores aceptan conexiones de todos los usuarios de los sistemas de las subredes 192.168.1.* y 10.0.0.*, excepto del sistema con la dirección 192.168.1.2. Todos los usuarios deben estar en el grupo **example-group**. El servidor OpenSSH rechaza todas las demás conexiones.

  Tenga en cuenta que el uso de listas de permitidos (directivas que empiezan por Allow) es más seguro que el uso de listas de bloqueados (opciones que empiezan por Deny) porque las listas de permitidos bloquean también a nuevos usuarios o grupos no autorizados.

Cambiar las políticas criptográficas de todo el sistema

- **OpenSSH** utiliza las políticas criptográficas de todo el sistema RHEL, y el nivel de política criptográfica por defecto de todo el sistema ofrece una configuración segura para los modelos de amenazas actuales. Para que la configuración criptográfica sea más estricta, cambie el nivel de política actual:

```
# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
```

- Para optar por las políticas de criptografía de todo el sistema para su servidor **OpenSSH**, descomente la línea con la variable **CRYPTO_POLICY=** en el archivo **/etc/sysconfig/sshd**. Después de este cambio, los valores que especifique en las secciones **Ciphers**, **MACs**, **KexAlgoritms**, y **GSSAPIKexAlgorithms** en el archivo **/etc/ssh/sshd_config** no serán anulados. Tenga en cuenta que esta tarea requiere una gran experiencia en la configuración de opciones criptográficas.

- Consulte Uso de políticas criptográficas en todo el sistema en el título de endurecimiento de la seguridad de RHEL 8 para obtener más información.

Recursos adicionales

- **sshd_config(5)**, **ssh-keygen(1)**, **crypto-policies(7)**, y **update-crypto-policies(8)** páginas de manual

1.6. CONECTARSE A UN SERVIDOR REMOTO UTILIZANDO UN HOST DE SALTO SSH

Utilice este procedimiento para conectarse a un servidor remoto a través de un servidor intermediario, también llamado host de salto.

Requisitos previos

- Un host de salto acepta conexiones SSH desde su sistema.

- Un servidor remoto acepta conexiones SSH sólo desde el host de salto.

Procedimiento

1. Defina el host de salto editando el archivo **~/.ssh/config**, por ejemplo:

```
Host jump-server1
  HostName jump1.example.com
```

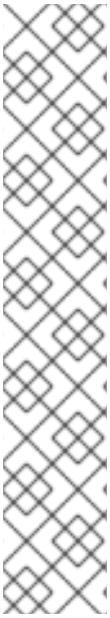
2. Añada la configuración de salto del servidor remoto con la directiva **ProxyJump** a **~/.ssh/config**, por ejemplo:

```
Host remote-server
  HostName remote1.example.com
  ProxyJump jump-server1
```

3. Conectar con el servidor remoto a través del servidor de salto:

```
$ ssh remote-server
```

El comando anterior es equivalente al comando **ssh -J jump-server1 remote-server** si se omiten los pasos de configuración 1 y 2.



NOTA

Puede especificar más servidores de salto y también puede omitir la adición de definiciones de host al archivo de configuraciones cuando proporciona sus nombres de host completos, por ejemplo:

```
$ ssh -J jump1.example.com,jump2.example.com,jump3.example.com
remote1.example.com
```

Cambie la notación de sólo nombre de host en el comando anterior si los nombres de usuario o los puertos SSH en los servidores de salto difieren de los nombres y puertos en el servidor remoto, por ejemplo:

```
$ ssh -J
johndoe@jump1.example.com:75,johndoe@jump2.example.com:75,johndoe@jump3.e
xample.com:75 joesec@remote1.example.com:220
```

Recursos adicionales

- **ssh_config(5)** y **ssh(1)** páginas man

1.7. CONEXIÓN A MÁQUINAS REMOTAS CON CLAVES SSH USANDO SSH-AGENT

Para evitar la introducción de una frase de contraseña cada vez que inicie una conexión SSH, puede utilizar la utilidad **ssh-agent** para almacenar en caché la clave privada SSH. La clave privada y la frase de contraseña permanecen seguras.

Requisitos previos

- Tienes un host remoto con el demonio SSH en ejecución y accesible a través de la red.
- Conoce la dirección IP o el nombre de host y las credenciales para iniciar sesión en el host remoto.
- Ha generado un par de claves SSH con una frase de paso y ha transferido la clave pública a la máquina remota. Para más información, consulte Generación de pares de claves SSH.

Procedimiento

1. Opcional: Compruebe que puede utilizar la clave para autenticarse en el host remoto:
   a. Conéctese al host remoto mediante SSH:

      ```
      $ ssh example.user1@198.51.100.1 hostname
      ```

   b. Introduzca la frase de contraseña que estableció al crear la clave para dar acceso a la clave privada.

```
$ ssh example.user1@198.51.100.1 hostname
 host.example.com
```

2. Inicie el **ssh-agent**.

```
$ eval $(ssh-agent)
Agent pid 20062
```

3. Añade la clave a **ssh-agent**.

```
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for ~/.ssh/id_rsa:
Identity added: ~/.ssh/id_rsa (example.user0@198.51.100.12)
```

Pasos de verificación

- Opcional: Inicie sesión en el equipo anfitrión mediante SSH.

```
$ ssh example.user1@198.51.100.1

Last login: Mon Sep 14 12:56:37 2020
```

Tenga en cuenta que no ha tenido que introducir la frase de contraseña.

1.8. RECURSOS ADICIONALES

Para obtener más información sobre la configuración y la conexión a los servidores y clientes de **OpenSSH** en Red Hat Enterprise Linux, consulte los recursos enumerados a continuación.

Documentación instalada

- **sshd(8)** La página de manual documenta las opciones disponibles en la línea de comandos y proporciona una lista completa de los archivos y directorios de configuración compatibles.
- la página de manual**ssh(1)** proporciona una lista completa de las opciones disponibles en la línea de comandos y de los archivos y directorios de configuración admitidos.
- la página de manual**scp(1)** proporciona una descripción más detallada de la utilidad **scp** y su uso.
- la página de manual**sftp(1)** proporciona una descripción más detallada de la utilidad **sftp** y su uso.
- la página man**ssh-keygen(1)** documenta en detalle el uso de la utilidad **ssh-keygen** para generar, gestionar y convertir las claves de autenticación utilizadas por ssh.
- la página de manual**ssh-copy-id(1)** describe el uso del script **ssh-copy-id**.
- **ssh_config(5)** La página de manual documenta las opciones de configuración del cliente SSH disponibles.
- la página de manual**sshd_config(5)** proporciona una descripción completa de las opciones de configuración disponibles del demonio SSH.

- la página de manual**update-crypto-policies(8)** proporciona orientación sobre la gestión de las políticas criptográficas de todo el sistema
- la página de manual**crypto-policies(7)** proporciona una visión general de los niveles de política criptográfica de todo el sistema

Documentación en línea

- Página principal de OpenSSH: contiene más documentación, preguntas frecuentes, enlaces a las listas de correo, informes de errores y otros recursos útiles.
- Configuración de SELinux para aplicaciones y servicios con configuraciones no estándar: puede aplicar procedimientos análogos para OpenSSH en una configuración no estándar con SELinux en modo de refuerzo.
- Controlar el tráfico de la red utilizando firewalld - proporciona orientación sobre la actualización de la configuración de **firewalld** después de cambiar un puerto SSH

CAPÍTULO 2. PLANIFICACIÓN Y APLICACIÓN DE TLS

TLS (Transport Layer Security) es un protocolo criptográfico utilizado para asegurar las comunicaciones de red. A la hora de endurecer los ajustes de seguridad del sistema configurando los protocolos de intercambio de claves, los métodos de autenticación y los algoritmos de cifrado preferidos, es necesario tener en cuenta que cuanto más amplia sea la gama de clientes admitidos, menor será la seguridad resultante. A la inversa, una configuración de seguridad estricta conlleva una compatibilidad limitada con los clientes, lo que puede provocar que algunos usuarios se queden fuera del sistema. Asegúrese de apuntar a la configuración más estricta disponible y sólo relájela cuando sea necesario por razones de compatibilidad.

2.1. PROTOCOLOS SSL Y TLS

El protocolo Secure Sockets Layer (SSL) fue desarrollado originalmente por Netscape Corporation para proporcionar un mecanismo de comunicación segura en Internet. Posteriormente, el protocolo fue adoptado por el Grupo de Trabajo de Ingeniería de Internet (IETF) y rebautizado como Transport Layer Security (TLS).

El protocolo TLS se sitúa entre una capa de protocolo de aplicación y una capa de transporte fiable, como TCP/IP. Es independiente del protocolo de aplicación y, por lo tanto, puede colocarse por debajo de muchos protocolos diferentes, por ejemplo: HTTP, FTP, SMTP, etc.

Versión del protocolo	Recomendación de uso
SSL v2	No utilizar. Tiene graves vulnerabilidades de seguridad. Eliminado de las bibliotecas criptográficas del núcleo desde RHEL 7.
SSL v3	No utilizar. Tiene graves vulnerabilidades de seguridad. Eliminado de las bibliotecas criptográficas del núcleo desde RHEL 8.
TLS 1.0	No se recomienda su uso. Tiene problemas conocidos que no se pueden mitigar de forma que se garantice la interoperabilidad, y no admite suites de cifrado modernas. Habilitado sólo en el perfil de política criptográfica de todo el sistema **LEGACY**.
TLS 1.1	Utilícelo con fines de interoperabilidad cuando sea necesario. No admite suites de cifrado modernas. Habilitado sólo en la política **LEGACY**.
TLS 1.2	Soporta las modernas suites de cifrado AEAD. Esta versión está habilitada en todas las políticas criptográficas del sistema, pero las partes opcionales de este protocolo contienen vulnerabilidades y TLS 1.2 también permite algoritmos obsoletos.
TLS 1.3	Versión recomendada. TLS 1.3 elimina las opciones problemáticas conocidas, proporciona privacidad adicional al cifrar una mayor parte del apretón de manos de la negociación y puede ser más rápido gracias al uso de algoritmos criptográficos modernos más eficientes. TLS 1.3 también está habilitado en todas las políticas criptográficas del sistema.

Recursos adicionales

- IETF: Protocolo de seguridad de la capa de transporte (TLS) versión 1.3

2.2. CONSIDERACIONES DE SEGURIDAD PARA TLS EN RHEL 8

En RHEL 8, las consideraciones relacionadas con la criptografía se simplifican significativamente gracias a las políticas de criptografía de todo el sistema. La política criptográfica **DEFAULT** sólo permite TLS 1.2 y 1.3. Para permitir que su sistema negocie conexiones utilizando las versiones anteriores de TLS, debe optar por no seguir las políticas criptográficas en una aplicación o cambiar a la política **LEGACY** con el comando **update-crypto-policies**. Consulte Uso de políticas criptográficas en todo el sistema para obtener más información.

La configuración por defecto proporcionada por las bibliotecas incluidas en RHEL 8 es lo suficientemente segura para la mayoría de las implementaciones. Las implementaciones de TLS utilizan algoritmos seguros siempre que sea posible, sin impedir las conexiones desde o hacia clientes o servidores heredados. Aplique configuraciones reforzadas en entornos con requisitos de seguridad estrictos en los que no se espera ni se permite la conexión de clientes o servidores heredados que no soportan algoritmos o protocolos seguros.

La forma más directa de endurecer la configuración de TLS es cambiar el nivel de la política criptográfica de todo el sistema a **FUTURE** utilizando el comando **update-crypto-policies --set FUTURE**.

Si decide no seguir las políticas de cifrado de todo el sistema RHEL, utilice las siguientes recomendaciones para los protocolos, conjuntos de cifrado y longitudes de clave preferidos en su configuración personalizada:

2.2.1. Protocolos

La última versión de TLS proporciona el mejor mecanismo de seguridad. A menos que tenga una razón de peso para incluir el soporte de versiones anteriores de TLS, permita que sus sistemas negocien las conexiones utilizando al menos la versión 1.2 de TLS. Tenga en cuenta que, a pesar de que RHEL 8 soporta la versión 1.3 de TLS, no todas las características de este protocolo son totalmente compatibles con los componentes de RHEL 8. Por ejemplo, la función 0-RTT (Zero Round Trip Time), que reduce la latencia de la conexión, todavía no está totalmente soportada por los servidores web Apache o Nginx.

2.2.2. Suites de cifrado

Las suites de cifrado modernas y más seguras deben preferirse a las antiguas e inseguras. Desactive siempre el uso de las suites de cifrado eNULL y aNULL, que no ofrecen ningún tipo de cifrado o autenticación. Si es posible, las suites de cifrado basadas en RC4 o HMAC-MD5, que tienen serias deficiencias, también deberían deshabilitarse. Lo mismo se aplica a las llamadas suites de cifrado de exportación, que se han hecho intencionadamente más débiles y, por tanto, son fáciles de romper.

Aunque no son inmediatamente inseguras, las suites de cifrado que ofrecen menos de 128 bits de seguridad no deberían considerarse por su corta vida útil. Los algoritmos que utilizan 128 bits de seguridad o más pueden esperarse que sean indescifrables durante al menos varios años, por lo que se recomiendan encarecidamente. Tenga en cuenta que aunque los cifrados 3DES anuncian el uso de 168 bits, en realidad ofrecen 112 bits de seguridad.

Siempre hay que dar preferencia a las suites de cifrado que soportan el secreto (perfecto) hacia adelante (PFS), que garantiza la confidencialidad de los datos cifrados incluso en caso de que la clave del servidor se vea comprometida. Esto descarta el rápido intercambio de claves RSA, pero permite el uso de ECDHE y DHE. De los dos, ECDHE es el más rápido y, por tanto, la opción preferida.

También debe dar preferencia a los cifrados AEAD, como AES-GCM, antes que a los cifrados en modo CBC, ya que no son vulnerables a los ataques de oráculo de relleno. Además, en muchos casos, AES-GCM es más rápido que AES en modo CBC, especialmente cuando el hardware tiene aceleradores criptográficos para AES.

Tenga en cuenta también que cuando se utiliza el intercambio de claves ECDHE con certificados ECDSA, la transacción es incluso más rápida que el intercambio de claves RSA puro. Para dar soporte a los clientes antiguos, puedes instalar dos pares de certificados y claves en un servidor: uno con claves ECDSA (para los nuevos clientes) y otro con claves RSA (para los antiguos).

2.2.3. Longitud de la clave pública

Cuando utilice claves RSA, prefiera siempre longitudes de clave de al menos 3072 bits firmadas por al menos SHA-256, que es lo suficientemente grande para una seguridad real de 128 bits.

AVISO

La seguridad de tu sistema es tan fuerte como el eslabón más débil de la cadena. Por ejemplo, un cifrado fuerte por sí solo no garantiza una buena seguridad. Las claves y los certificados son igual de importantes, así como las funciones hash y las claves utilizadas por la Autoridad de Certificación (CA) para firmar sus claves.

Recursos adicionales

- Políticas criptográficas para todo el sistema en RHEL 8 .
- **update-crypto-policies(8)** página de manual

2.3. ENDURECIMIENTO DE LA CONFIGURACIÓN DE TLS EN LAS APLICACIONES

En Red Hat Enterprise Linux 8, las políticas criptográficas de todo el sistema proporcionan una manera conveniente de asegurar que sus aplicaciones que utilizan bibliotecas criptográficas no permiten protocolos, cifrados o algoritmos inseguros conocidos.

Si desea endurecer su configuración relacionada con TLS con sus ajustes criptográficos personalizados, puede utilizar las opciones de configuración criptográfica descritas en esta sección, y anular las políticas criptográficas de todo el sistema sólo en la cantidad mínima requerida.

Independientemente de la configuración que elija utilizar, asegúrese siempre de que su aplicación de servidor aplique *server-side cipher order* , de modo que el conjunto de cifrado que se utilice esté determinado por la orden que configure.

2.3.1. Configuración de la Apache HTTP server

El **Apache HTTP Server** puede utilizar las bibliotecas **OpenSSL** y **NSS** para sus necesidades de TLS. Red Hat Enterprise Linux 8 proporciona la funcionalidad de **mod_ssl** a través de paquetes epónimos:

```
# yum install mod_ssl
```

El paquete **mod_ssl** instala el archivo de configuración **/etc/httpd/conf.d/ssl.conf**, que puede utilizarse para modificar los ajustes relacionados con TLS de **Apache HTTP Server**.

Instale el paquete **httpd-manual** para obtener la documentación completa de **Apache HTTP Server**,

incluida la configuración de TLS. Las directivas disponibles en el archivo de configuración **/etc/httpd/conf.d/ssl.conf** se describen en detalle en /usr/share/httpd/manual/mod/mod_ssl.html. Ejemplos de varias configuraciones están en /usr/share/httpd/manual/ssl/ssl_howto.html.

Al modificar los ajustes en el archivo de configuración **/etc/httpd/conf.d/ssl.conf**, asegúrese de tener en cuenta como mínimo las tres directivas siguientes:

SSLProtocol

Utilice esta directiva para especificar la versión de TLS o SSL que desea permitir.

SSLCipherSuite

Utilice esta directiva para especificar su conjunto de cifrado preferido o deshabilitar los que desee no permitir.

SSLHonorCipherOrder

Descomente y establezca esta directiva en **on** para asegurarse de que los clientes que se conectan se adhieren al orden de cifrado que ha especificado.

Por ejemplo, para utilizar sólo el protocolo TLS 1.2 y 1.3:

```
SSLProtocol             all -SSLv3 -TLSv1 -TLSv1.1
```

2.3.2. Configuración del servidor HTTP y proxy de Nginx

Para habilitar la compatibilidad con TLS 1.3 en **Nginx**, añada el valor **TLSv1.3** a la opción **ssl_protocols** en la sección **server** del archivo de configuración **/etc/nginx/nginx.conf**:

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    ....
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers
    ....
}
```

2.3.3. Configuración del servidor de correo Dovecot

Para configurar su instalación del servidor de correo **Dovecot** para utilizar TLS, modifique el archivo de configuración **/etc/dovecot/conf.d/10-ssl.conf**. Puede encontrar una explicación de algunas de las directivas de configuración básicas disponibles en ese archivo en el archivo /usr/share/doc/dovecot/wiki/SSL.DovecotConfiguration.txt, que se instala junto con la instalación estándar de **Dovecot**.

Al modificar los ajustes en el archivo de configuración **/etc/dovecot/conf.d/10-ssl.conf**, asegúrese de tener en cuenta como mínimo las tres directivas siguientes:

ssl_protocols

Utilice esta directiva para especificar la versión de TLS o SSL que desea permitir o deshabilitar.

ssl_cipher_list

Utilice esta directiva para especificar sus suites de cifrado preferidas o deshabilitar las que desee no permitir.

ssl_prefer_server_ciphers

Descomente y establezca esta directiva en **yes** para asegurarse de que los clientes que se conectan se adhieren al orden de cifrado que ha especificado.

Por ejemplo, la siguiente línea en **/etc/dovecot/conf.d/10-ssl.conf** sólo permite TLS 1.1 y posteriores:

```
ssl_protocols = !SSLv2 !SSLv3 !TLSv1
```

Recursos adicionales

Para obtener más información sobre la configuración de TLS y otros temas relacionados, consulte los recursos que se indican a continuación.

- la página de manual**config(5)** describe el formato del archivo de configuración **/etc/ssl/openssl.conf**.
- la página de manual**ciphers(1)** incluye una lista de palabras clave y cadenas de cifrado disponibles en **OpenSSL**.
- Recomendaciones para el uso seguro de la seguridad de la capa de transporte (TLS) y la seguridad de la capa de transporte de datagramas (DTLS)
- El generador de configuración de Mozilla SSL puede ayudar a crear archivos de configuración para **Apache** o **Nginx** con configuraciones seguras que deshabilitan los protocolos, cifrados y algoritmos hash vulnerables conocidos.
- Laprueba de servidor SSL verifica que su configuración cumple con los requisitos de seguridad modernos.

CAPÍTULO 3. CONFIGURACIÓN DE UNA VPN CON IPSEC

En Red Hat Enterprise Linux 8, se puede configurar una red privada virtual (VPN) utilizando el protocolo **IPsec**, que es soportado por la aplicación **Libreswan**.

3.1. LIBRESWAN COMO IMPLEMENTACIÓN DE VPN IPSEC

En Red Hat Enterprise Linux 8, se puede configurar una Red Privada Virtual (VPN) utilizando el protocolo **IPsec**, que es soportado por la aplicación **Libreswan**. **Libreswan** es una continuación de la aplicación **Openswan**, y muchos ejemplos de la documentación **Openswan** son intercambiables con **Libreswan**.

El protocolo **IPsec** para una VPN se configura utilizando el protocolo Internet Key Exchange (**IKE**). Los términos IPsec e IKE se utilizan indistintamente. Una VPN IPsec también se denomina VPN IKE, VPN IKEv2, VPN XAUTH, VPN Cisco o VPN IKE/IPsec. Una variante de una VPN IPsec que también utiliza el Protocolo de Túnel de Nivel 2 (**L2TP**) suele llamarse VPN L2TP/IPsec, que requiere la aplicación del canal opcional **xl2tpd**.

Libreswan es una implementación de código abierto y espacio de usuario de **IKE**. **IKE** v1 y v2 se implementan como un demonio a nivel de usuario. El protocolo IKE también está cifrado. El protocolo **IPsec** es implementado por el kernel de Linux, y **Libreswan** configura el kernel para añadir y eliminar configuraciones de túneles VPN.

El protocolo **IKE** utiliza los puertos UDP 500 y 4500. El protocolo **IPsec** consta de dos protocolos:

- Encapsulated Security Payload (**ESP**), que tiene el número de protocolo 50.
- Authenticated Header (**AH**), que tiene el número de protocolo 51.

No se recomienda el uso del protocolo **AH**. Se recomienda a los usuarios de **AH** que migren a **ESP** con cifrado nulo.

El protocolo **IPsec** ofrece dos modos de funcionamiento:

- **Tunnel Mode** (por defecto)
- **Transport Mode**.

Se puede configurar el kernel con IPsec sin IKE. Esto se llama **Manual Keying**. También puede configurar la clave manual utilizando los comandos de **ip xfrm**, sin embargo, esto se desaconseja fuertemente por razones de seguridad. **Libreswan** interactúa con el kernel de Linux utilizando netlink. El cifrado y descifrado de paquetes se realiza en el kernel de Linux.

Libreswan utiliza la biblioteca criptográfica Network Security Services (**NSS**). Tanto **Libreswan** como **NSS** están certificados para su uso con la Publicación 140-2 de *Federal Information Processing Standard* (**FIPS**).

IMPORTANTE

IKE/**IPsec** VPN, implementada por **Libreswan** y el kernel de Linux, es la única tecnología VPN recomendada para su uso en Red Hat Enterprise Linux 8. No utilice ninguna otra tecnología VPN sin entender los riesgos de hacerlo.

En Red Hat Enterprise Linux 8, **Libreswan** sigue a **system-wide cryptographic policies** por defecto. Esto asegura que **Libreswan** utiliza configuraciones seguras para los modelos de amenazas actuales,

incluyendo **IKEv2** como protocolo por defecto. Consulte Uso de políticas criptográficas en todo el sistema para obtener más información.

Libreswan no utiliza los términos "origen" y "destino" o "servidor" y "cliente" porque IKE/IPsec son protocolos peer to peer. En su lugar, utiliza los términos "izquierda" y "derecha" para referirse a los puntos finales (los hosts). Esto también permite utilizar la misma configuración en ambos puntos finales en la mayoría de los casos. Sin embargo, los administradores suelen optar por utilizar siempre "izquierda" para el host local y "derecha" para el host remoto.

3.2. INSTALACIÓN DE LIBRESWAN

Este procedimiento describe los pasos para instalar e iniciar la implementación de la VPN IPsec/IKE de **Libreswan**.

Requisitos previos

- El repositorio **AppStream** está activado.

Procedimiento

1. Instale los paquetes de **libreswan**:

```
# yum install libreswan
```

2. Si está reinstalando **Libreswan**, elimine sus antiguos archivos de base de datos:

```
# systemctl stop ipsec
# rm /etc/ipsec.d/*db
```

3. Inicie el servicio **ipsec**, y habilite el servicio para que se inicie automáticamente al arrancar:

```
# systemctl enable ipsec --now
```

4. Configure el cortafuegos para permitir los puertos 500 y 4500/UDP para los protocolos IKE, ESP y AH añadiendo el servicio **ipsec**:

```
# firewall-cmd --add-service="ipsec"
# firewall-cmd --runtime-to-permanent
```

3.3. CREACIÓN DE UNA VPN DE HOST A HOST

Para configurar **Libreswan** para crear una VPN de host a host **IPsec** entre dos hosts denominados *left* y *right*, introduzca los siguientes comandos en ambos hosts:

Procedimiento

1. Generar un par de claves RSA en cada host:

```
# ipsec newhostkey --output /etc/ipsec.d/hostkey.secrets
```

2. El paso anterior devolvió la clave generada **ckaid**. Utilice ese **ckaid** con el siguiente comando en *left*, por ejemplo:

```
# ipsec showhostkey --left --ckaid 2d3ea57b61c9419dfd6cf43a1eb6cb306c0e857d
```

La salida del comando anterior generó la línea **leftrsasigkey=** necesaria para la configuración. Haga lo mismo en el segundo host (*right*):

```
# ipsec showhostkey --right --ckaid a9e1f6ce9ecd3608c24e8f701318383f41798f03
```

3. En el directorio **/etc/ipsec.d/**, cree un nuevo archivo **my_host-to-host.conf**. Escriba en el nuevo archivo las claves de host RSA de la salida de los comandos de **ipsec showhostkey** en el paso anterior. Por ejemplo:

```
conn mytunnel
    leftid=@west
    left=192.1.2.23
    leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
    rightid=@east
    right=192.1.2.45
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    authby=rsasig
```

4. Después de importar las claves, reinicie el servicio **ipsec**:

```
# systemctl restart ipsec
```

5. Inicio **Libreswan**:

```
# ipsec setup start
```

6. Cargue la conexión:

```
# ipsec auto --add mytunnel
```

7. Establece el túnel:

```
# ipsec auto --up mytunnel
```

8. Para iniciar automáticamente el túnel cuando se inicie el servicio **ipsec**, añada la siguiente línea a la definición de la conexión:

```
auto=inicio
```

3.4. CONFIGURACIÓN DE UNA VPN DE SITIO A SITIO

Para crear una VPN de sitio a sitio **IPsec**, al unir dos redes, se crea un túnel **IPsec** entre los dos hosts. Los hosts actúan así como puntos finales, que están configurados para permitir el paso del tráfico de una o más subredes. Por lo tanto, se puede pensar en los hosts como puertas de enlace hacia la parte remota de la red.

La configuración de la VPN de sitio a sitio sólo difiere de la VPN de host a host en que hay que especificar una o más redes o subredes en el archivo de configuración.

Requisitos previos

- Una VPN de host a host ya está configurada.

Procedimiento

1. Copie el archivo con la configuración de su VPN de host a host en un nuevo archivo, por ejemplo:

```
# cp /etc/ipsec.d/my_host-to-host.conf /etc/ipsec.d/my_site-to-site.conf
```

2. Añada la configuración de la subred al archivo creado en el paso anterior, por ejemplo:

```
conn mysubnet
    also=mytunnel
    leftsubnet=192.0.1.0/24
    rightsubnet=192.0.2.0/24
    auto=start

conn mysubnet6
    also=mytunnel
    leftsubnet=2001:db8:0:1::/64
    rightsubnet=2001:db8:0:2::/64
    auto=start

# the following part of the configuration file is the same for both host-to-host and site-to-site
connections:

conn mytunnel
   leftid=@west
   left=192.1.2.23
   leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
   rightid=@east
   right=192.1.2.45
   rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
   authby=rsasig
```

3.5. CONFIGURAR UNA VPN DE ACCESO REMOTO

Los guerreros de la carretera son usuarios que viajan con clientes móviles con una dirección IP asignada dinámicamente, como los ordenadores portátiles. Los clientes móviles se autentican mediante certificados.

El siguiente ejemplo muestra la configuración para **IKEv2**, y evita el uso del protocolo **IKEv1** XAUTH.

En el servidor:

```
conn roadwarriors
   ikev2=insist
   # Support (roaming) MOBIKE clients (RFC 4555)
   mobike=yes
   fragmentation=yes
   left=1.2.3.4
   # if access to the LAN is given, enable this, otherwise use 0.0.0.0/0
   # leftsubnet=10.10.0.0/16
   leftsubnet=0.0.0.0/0
```

```
    leftcert=gw.example.com
    leftid=%fromcert
    leftxauthserver=yes
    leftmodecfgserver=yes
    right=%any
    # trust our own Certificate Agency
    rightca=%same
    # pick an IP address pool to assign to remote users
    # 100.64.0.0/16 prevents RFC1918 clashes when remote users are behind NAT
    rightaddresspool=100.64.13.100-100.64.13.254
    # if you want remote clients to use some local DNS zones and servers
    modecfgdns="1.2.3.4, 5.6.7.8"
    modecfgdomains="internal.company.com, corp"
    rightxauthclient=yes
    rightmodecfgclient=yes
    authby=rsasig
    # optionally, run the client X.509 ID through pam to allow/deny client
    # pam-authorize=yes
    # load connection, don't initiate
    auto=add
    # kill vanished roadwarriors
    dpddelay=1m
    dpdtimeout=5m
    dpdaction=clear
```

En el cliente móvil, el dispositivo del guerrero de la carretera, utilice una ligera variación de la configuración anterior:

```
conn to-vpn-server
    ikev2=insist
    # pick up our dynamic IP
    left=%defaultroute
    leftsubnet=0.0.0.0/0
    leftcert=myname.example.com
    leftid=%fromcert
    leftmodecfgclient=yes
    # right can also be a DNS hostname
    right=1.2.3.4
    # if access to the remote LAN is required, enable this, otherwise use 0.0.0.0/0
    # rightsubnet=10.10.0.0/16
    rightsubnet=0.0.0.0/0
    fragmentation=yes
    # trust our own Certificate Agency
    rightca=%same
    authby=rsasig
    # allow narrowing to the server's suggested assigned IP and remote subnet
    narrowing=yes
    # Support (roaming) MOBIKE clients (RFC 4555)
    mobike=yes
    # Initiate connection
    auto=start
```

3.6. CONFIGURAR UNA VPN DE MALLA

Una red VPN en malla, que también se conoce como VPN *any-to-any*, es una red en la que todos los nodos se comunican utilizando **IPsec**. La configuración permite excepciones para los nodos que no pueden utilizar **IPsec**. La red VPN en malla puede configurarse de dos maneras:

- Requerir **IPsec**.
- Para preferir **IPsec**, pero permitir una vuelta a la comunicación en texto claro.

La autenticación entre los nodos puede basarse en certificados X.509 o en extensiones de seguridad DNS (DNSSEC).

El siguiente procedimiento utiliza certificados X.509. Estos certificados pueden generarse utilizando cualquier tipo de sistema de gestión de autoridades de certificación (CA), como el sistema de certificados Dogtag. Dogtag asume que los certificados de cada nodo están disponibles en el formato PKCS #12 (archivos .p12), que contienen la clave privada, el certificado del nodo y el certificado de la CA Raíz utilizado para validar los certificados X.509 de otros nodos.

Cada nodo tiene una configuración idéntica con la excepción de su certificado X.509. Esto permite añadir nuevos nodos sin reconfigurar ninguno de los existentes en la red. Los archivos PKCS #12 requieren un "nombre amistoso", para el que utilizamos el nombre "nodo", de modo que los archivos de configuración que hacen referencia al nombre amistoso pueden ser idénticos para todos los nodos.

Requisitos previos

- se instala**Libreswan** y se inicia el servicio **ipsec** en cada nodo.

Procedimiento

1. En cada nodo, importe los archivos PKCS #12. Este paso requiere la contraseña utilizada para generar los archivos PKCS #12:

```
# ipsec import nodeXXX.p12
```

2. Cree las siguientes tres definiciones de conexión para los perfiles **IPsec required** (privado), **IPsec optional** (privado o claro) y **No IPsec** (claro):

```
# cat /etc/ipsec.d/mesh.conf
conn clear
 auto=ondemand
 type=passthrough
 authby=never
 left=%defaultroute
 right=%group

conn private
 auto=ondemand
 type=transport
 authby=rsasig
 failureshunt=drop
 negotiationshunt=drop
 # left
 left=%defaultroute
 leftcert=nodeXXXX
 leftid=%fromcert
       leftrsasigkey=%cert
 # right
```

```
 rightrsasigkey=%cert
 rightid=%fromcert
 right=%opportunisticgroup

conn private-or-clear
 auto=ondemand
 type=transport
 authby=rsasig
 failureshunt=passthrough
 negotiationshunt=passthrough
 # left
 left=%defaultroute
 leftcert=nodeXXXX
 leftid=%fromcert
       leftrsasigkey=%cert
 # right
 rightrsasigkey=%cert
 rightid=%fromcert
 right=%opportunisticgroup
```

3. Añada la dirección IP de la red en la categoría adecuada. Por ejemplo, si todos los nodos residen en la red 10.15.0.0/16, y todos los nodos deben ordenar el cifrado **IPsec**:

```
# echo "10.15.0.0/16" >> /etc/ipsec.d/policies/private
```

4. Para permitir que ciertos nodos, por ejemplo, 10.15.34.0/24, trabajen con y sin **IPsec**, añada esos nodos al grupo de privados o limpios mediante:

```
# echo "10.15.34.0/24" >> /etc/ipsec.d/policies/private-or-clear
```

5. Para definir un host, por ejemplo, 10.15.1.2, que no es capaz de **IPsec** en el grupo claro, utilice:

```
# echo "10.15.1.2/32" >> /etc/ipsec.d/policies/clear
```

   Los archivos del directorio **/etc/ipsec.d/policies** se pueden crear a partir de una plantilla para cada nuevo nodo, o se pueden aprovisionar utilizando Puppet o Ansible.

   Tenga en cuenta que cada nodo tiene la misma lista de excepciones o diferentes expectativas de flujo de tráfico. Por lo tanto, dos nodos podrían no ser capaces de comunicarse porque uno requiere **IPsec** y el otro no puede utilizar **IPsec**.

6. Reinicie el nodo para añadirlo a la malla configurada:

```
# systemctl restart ipsec
```

7. Una vez que haya terminado con la adición de nodos, un comando **ping** es suficiente para abrir un túnel **IPsec**. Para ver qué túneles ha abierto un nodo:

```
# ipsec trafficstatus
```

3.7. MÉTODOS DE AUTENTIFICACIÓN UTILIZADOS EN LIBRESWAN

Puede utilizar los siguientes métodos para la autenticación de los puntos finales:

- *Pre-Shared Keys* (PSK) es el método de autenticación más sencillo. Los PSK deben estar formados por caracteres aleatorios y tener una longitud de al menos 20 caracteres. En el modo FIPS, los PSKs deben cumplir con un requisito de fuerza mínima dependiendo del algoritmo de integridad utilizado. Se recomienda no utilizar PSKs de menos de 64 caracteres aleatorios.

- *Raw RSA keys* se utiliza habitualmente para configuraciones estáticas de host a host o de subred a subred **IPsec**. Los hosts se configuran manualmente con la clave RSA pública de cada uno. Este método no se adapta bien cuando docenas o más hosts necesitan configurar túneles **IPsec** entre sí.

- *X.509 certificates* se utiliza habitualmente para despliegues a gran escala en los que hay muchos hosts que necesitan conectarse a una pasarela común **IPsec**. Se utiliza una *certificate authority* (CA) central para firmar certificados RSA para hosts o usuarios. Esta CA central es responsable de transmitir la confianza, incluyendo las revocaciones de hosts o usuarios individuales.

- *NULL authentication* se utiliza para obtener un cifrado de malla sin autenticación. Protege contra los ataques pasivos pero no protege contra los ataques activos. Sin embargo, como **IKEv2** permite métodos de autenticación asimétricos, la autenticación NULL también puede utilizarse para IPsec oportunista a escala de Internet, donde los clientes autentican al servidor, pero los servidores no autentican al cliente. Este modelo es similar al de los sitios web seguros que utilizan **TLS**.

Protección contra los ordenadores cuánticos

Además de estos métodos de autenticación, puede utilizar el método *Postquantum Preshared Keys* (PPK) para protegerse de posibles ataques de ordenadores cuánticos. Los clientes individuales o los grupos de clientes pueden utilizar su propia PPK especificando un (PPKID) que corresponde a una clave precompartida configurada fuera de banda.

El uso de **IKEv1** con claves precompartidas ofrecía protección contra los atacantes cuánticos. El rediseño de **IKEv2** no ofrece esta protección de forma nativa. **Libreswan** ofrece el uso de *Postquantum Preshared Keys* (PPK) para proteger las conexiones de **IKEv2** contra los ataques cuánticos.

Para activar el soporte opcional de PPK, añada **ppk=yes** a la definición de la conexión. Para exigir la PPK, añada **ppk=insist**. Entonces, a cada cliente se le puede dar un ID de PPK con un valor secreto que se comunica fuera de banda (y preferiblemente seguro desde el punto de vista cuántico). Las PPK deben ser muy fuertes en aleatoriedad y no estar basadas en palabras de diccionario. El ID PPK y los datos PPK en sí se almacenan en **ipsec.secrets**, por ejemplo:

```
@west @east : PPKS \ "user1" \N - "la cuerda es para llevarla a cabo"
```

La opción **PPKS** se refiere a los PPK estáticos. Una función experimental utiliza PPKs dinámicos basados en una almohadilla de un solo uso. En cada conexión, se utiliza una nueva parte de una almohadilla de un solo uso como PPK. Cuando se utiliza, esa parte del PPK dinámico dentro del archivo se sobrescribe con ceros para evitar su reutilización. Si no queda más material de la almohadilla de un solo uso, la conexión falla. Consulte la página de manual **ipsec.secrets(5)** para obtener más información.

AVISO

La implementación de los PPK dinámicos se proporciona como una Muestra de Tecnología, y esta funcionalidad debe utilizarse con precaución.

3.8. IMPLEMENTACIÓN DE UNA VPN IPSEC COMPATIBLE CON FIPS

Utilice este procedimiento para implementar una solución VPN IPsec compatible con FIPS basada en Libreswan. Los siguientes pasos también le permiten identificar qué algoritmos criptográficos están disponibles y cuáles están desactivados para Libreswan en modo FIPS.

Requisitos previos

- El repositorio **AppStream** está activado.

Procedimiento

1. Instale los paquetes de **libreswan**:

```
# yum install libreswan
```

2. Si está reinstalando **Libreswan**, elimine su antigua base de datos NSS:

```
# systemctl stop ipsec
# rm /etc/ipsec.d/*db
```

3. Inicie el servicio **ipsec**, y habilite el servicio para que se inicie automáticamente al arrancar:

```
# systemctl enable ipsec --now
```

4. Configure el cortafuegos para permitir los puertos 500 y 4500/UDP para los protocolos IKE, ESP y AH añadiendo el servicio **ipsec**:

```
# firewall-cmd --add-service="ipsec"
# firewall-cmd --runtime-to-permanent
```

5. Cambie el sistema al modo FIPS en RHEL 8:

```
# fips-mode-setup --enable
```

6. Reinicie su sistema para permitir que el kernel cambie al modo FIPS:

```
# reboot
```

Pasos de verificación

1. Para confirmar que Libreswan funciona en modo FIPS:

```
# ipsec whack --fipsstatus
000 FIPS mode enabled
```

2. También puede comprobar las entradas de la unidad **ipsec** en el diario **systemd**:

```
$ journalctl -u ipsec
...
Jan 22 11:26:50 localhost.localdomain pluto[3076]: FIPS Product: YES
```

```
Jan 22 11:26:50 localhost.localdomain pluto[3076]: FIPS Kernel: YES
Jan 22 11:26:50 localhost.localdomain pluto[3076]: FIPS Mode: YES
```

3. Para ver los algoritmos disponibles en el modo FIPS:

```
# ipsec pluto --selftest 2>&1 | head -11
FIPS Product: YES
FIPS Kernel: YES
FIPS Mode: YES
NSS DB directory: sql:/etc/ipsec.d
Initializing NSS
Opening NSS database "sql:/etc/ipsec.d" read-only
NSS initialized
NSS crypto library initialized
FIPS HMAC integrity support [enabled]
FIPS mode enabled for pluto daemon
NSS library is running in FIPS mode
FIPS HMAC integrity verification self-test passed
```

4. Para consultar los algoritmos desactivados en el modo FIPS:

```
# ipsec pluto --selftest 2>&1 | grep disabled
Encryption algorithm CAMELLIA_CTR disabled; not FIPS compliant
Encryption algorithm CAMELLIA_CBC disabled; not FIPS compliant
Encryption algorithm SERPENT_CBC disabled; not FIPS compliant
Encryption algorithm TWOFISH_CBC disabled; not FIPS compliant
Encryption algorithm TWOFISH_SSH disabled; not FIPS compliant
Encryption algorithm NULL disabled; not FIPS compliant
Encryption algorithm CHACHA20_POLY1305 disabled; not FIPS compliant
Hash algorithm MD5 disabled; not FIPS compliant
PRF algorithm HMAC_MD5 disabled; not FIPS compliant
PRF algorithm AES_XCBC disabled; not FIPS compliant
Integrity algorithm HMAC_MD5_96 disabled; not FIPS compliant
Integrity algorithm HMAC_SHA2_256_TRUNCBUG disabled; not FIPS compliant
Integrity algorithm AES_XCBC_96 disabled; not FIPS compliant
DH algorithm MODP1024 disabled; not FIPS compliant
DH algorithm MODP1536 disabled; not FIPS compliant
DH algorithm DH31 disabled; not FIPS compliant
```

5. Para listar todos los algoritmos y cifrados permitidos en el modo FIPS:

```
# ipsec pluto --selftest 2>&1 | grep ESP | grep FIPS | sed "s/^.*FIPS//"
{256,192,*128}  aes_ccm, aes_ccm_c
{256,192,*128}  aes_ccm_b
{256,192,*128}  aes_ccm_a
[*192]  3des
{256,192,*128}  aes_gcm, aes_gcm_c
{256,192,*128}  aes_gcm_b
{256,192,*128}  aes_gcm_a
{256,192,*128}  aesctr
{256,192,*128}  aes
{256,192,*128}  aes_gmac
sha, sha1, sha1_96, hmac_sha1
sha512, sha2_512, sha2_512_256, hmac_sha2_512
sha384, sha2_384, sha2_384_192, hmac_sha2_384
```

```
sha2, sha256, sha2_256, sha2_256_128, hmac_sha2_256
aes_cmac
null
null, dh0
dh14
dh15
dh16
dh17
dh18
ecp_256, ecp256
ecp_384, ecp384
ecp_521, ecp521
```

Recursos adicionales

- Uso de políticas criptográficas en todo el sistema

3.9. PROTEGER LA BASE DE DATOS IPSEC NSS CON UNA CONTRASEÑA

Por defecto, el servicio IPsec crea su base de datos de Servicios de Seguridad de Red (NSS) con una contraseña vacía durante el primer inicio. Añada la protección con contraseña siguiendo los siguientes pasos.

NOTA

En las versiones anteriores de RHEL hasta la versión 6.6, había que proteger la base de datos de IPsec NSS con una contraseña para cumplir los requisitos de FIPS 140-2 porque las bibliotecas criptográficas de NSS estaban certificadas para el estándar FIPS 140-2 Nivel 2. En RHEL 8, el NIST certificó NSS para el nivel 1 de esta norma, y este estado no requiere la protección con contraseña de la base de datos.

Requisito previo

- El directorio **/etc/ipsec.d** contiene los archivos de la base de datos NSS.

Procedimiento

1. Activar la protección por contraseña de la base de datos **NSS** para **Libreswan**:

```
# certutil -N -d sql:/etc/ipsec.d
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
```

2. Cree el archivo **/etc/ipsec.d/nsspassword** con la contraseña que ha establecido en el paso anterior, por ejemplo:

```
# cat /etc/ipsec.d/nsspassword
NSS Certificate DB:MyStrongPasswordHere
```

Tenga en cuenta que el archivo **nsspassword** utiliza la siguiente sintaxis:

```
token_1_name:the_password
token_2_name:the_password
```

El token de software NSS por defecto es **NSS Certificate DB**. Si su sistema funciona en modo FIPS, el nombre del token es **NSS FIPS 140-2 Certificate DB**.

3. Dependiendo de su escenario, inicie o reinicie el servicio **ipsec** después de terminar el archivo **nsspassword**:

```
# systemctl restart ipsec
```

Pasos de verificación

1. Compruebe que el servicio **ipsec** está funcionando después de haber añadido una contraseña no vacía a su base de datos NSS:

```
# systemctl status ipsec
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
  Loaded: loaded (/usr/lib/systemd/system/ipsec.service; enabled; vendor preset: disable>
  Active: active (running)...
```

2. Opcionalmente, compruebe que el registro **Journal** contiene entradas que confirman una inicialización exitosa:

```
# journalctl -u ipsec
...
pluto[23001]: NSS DB directory: sql:/etc/ipsec.d
pluto[23001]: Initializing NSS
pluto[23001]: Opening NSS database "sql:/etc/ipsec.d" read-only
pluto[23001]: NSS Password from file "/etc/ipsec.d/nsspassword" for token "NSS Certificate
DB" with length 20 passed to NSS
pluto[23001]: NSS crypto library initialized
...
```

Recursos adicionales

- La página de manual **certutil(1)**.
- Para obtener más información sobre las certificaciones relacionadas con FIPS 140-2, consulte el artículo de la base de conocimientos sobre normas gubernamentales.

3.10. CONFIGURAR LAS CONEXIONES IPSEC QUE OPTAN POR LAS POLÍTICAS CRIPTOGRÁFICAS DE TODO EL SISTEMA

Anulación de las políticas criptográficas de todo el sistema para una conexión

Las políticas criptográficas de todo el sistema RHEL crean una conexión especial llamada **fault**. Esta conexión contiene los valores por defecto para las opciones **ikev2**, **esp**, y **ike**. Sin embargo, puede anular los valores por defecto especificando la opción mencionada en el archivo de configuración de la conexión.

Por ejemplo, la siguiente configuración permite conexiones que utilizan IKEv1 con AES y SHA-1 o SHA-2, e IPsec (ESP) con AES-GCM o AES-CBC:

```
conn MyExample
  ...
  ikev2=never
  ike=aes-sha2,aes-sha1;modp2048
  esp=aes_gcm,aes-sha2,aes-sha1
  ...
```

Tenga en cuenta que AES-GCM está disponible para IPsec (ESP) y para IKEv2, pero no para IKEv1.

Desactivación de las políticas criptográficas de todo el sistema para todas las conexiones

Para desactivar las políticas criptográficas de todo el sistema para todas las conexiones IPsec, comente la siguiente línea en el archivo **/etc/ipsec.conf**:

```
incluir /etc/crypto-policies/back-ends/libreswan.config
```

A continuación, añada la opción **ikev2=never** a su archivo de configuración de la conexión.

Recursos adicionales

- Para más información, consulte Uso de políticas criptográficas en todo el sistema .

3.11. RESOLUCIÓN DE PROBLEMAS DE CONFIGURACIÓN DE VPN IPSEC

Los problemas relacionados con las configuraciones de VPN IPsec suelen producirse por varias razones principales. Si se encuentra con este tipo de problemas, puede comprobar si la causa del problema se corresponde con alguno de los siguientes escenarios, y aplicar la solución correspondiente.

Solución de problemas básicos de conexión

La mayoría de los problemas con las conexiones VPN se producen en las nuevas implantaciones, en las que los administradores configuran los puntos finales con opciones de configuración que no coinciden. Además, una configuración que funciona puede dejar de hacerlo repentinamente, a menudo debido a valores incompatibles introducidos recientemente. Esto puede ser el resultado de que un administrador cambie la configuración. Alternativamente, un administrador puede haber instalado una actualización de firmware o una actualización de paquete con diferentes valores por defecto para ciertas opciones, como los algoritmos de cifrado.

Para confirmar que se ha establecido una conexión VPN IPsec:

```
# ipsec trafficstatus
006 #8: "vpn.example.com"[1] 192.0.2.1, type=ESP, add_time=1595296930, inBytes=5999,
outBytes=3231, id='@vpn.example.com', lease=100.64.13.5/32
```

Si la salida está vacía o no muestra una entrada con el nombre de la conexión, el túnel está roto.

Para comprobar que el problema está en la conexión:

1. Vuelva a cargar la conexión *vpn.example.com*:

```
# ipsec auto --add vpn.example.com
002 added connection description "vpn.example.com"
```

2. A continuación, inicie la conexión VPN:

```
# ipsec auto --up vpn.example.com
```

Problemas relacionados con los cortafuegos

El problema más común es que un firewall en uno de los puntos finales de IPsec o en un router entre los puntos finales está dejando caer todos los paquetes de intercambio de claves de Internet (IKE).

- Para IKEv2, una salida similar al siguiente ejemplo indica un problema con un firewall:

```
# ipsec auto --up vpn.example.com
181 "vpn.example.com"[1] 192.0.2.2 #15: initiating IKEv2 IKE SA
181 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: sent v2I1, expected v2R1
010 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: retransmission; will wait 0.5
seconds for response
010 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: retransmission; will wait 1
seconds for response
010 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: retransmission; will wait 2
seconds for
...
```

- Para IKEv1, la salida del comando de iniciación tiene el siguiente aspecto:

```
# ipsec auto --up vpn.example.com
002 "vpn.example.com" #9: initiating Main Mode
102 "vpn.example.com" #9: STATE_MAIN_I1: sent MI1, expecting MR1
010 "vpn.example.com" #9: STATE_MAIN_I1: retransmission; will wait 0.5 seconds for
response
010 "vpn.example.com" #9: STATE_MAIN_I1: retransmission; will wait 1 seconds for
response
010 "vpn.example.com" #9: STATE_MAIN_I1: retransmission; will wait 2 seconds for
response
...
```

Debido a que el protocolo IKE, que se utiliza para configurar IPsec, está encriptado, sólo puede solucionar un subconjunto limitado de problemas utilizando la herramienta **tcpdump**. Si un cortafuegos está dejando caer paquetes IKE o IPsec, puedes intentar encontrar la causa utilizando la utilidad **tcpdump**. Sin embargo, **tcpdump** no puede diagnosticar otros problemas con las conexiones VPN IPsec.

- Para capturar la negociación de la VPN y todos los datos cifrados en la interfaz **eth0**:

```
# tcpdump -i eth0 -n -n esp or udp port 500 or udp port 4500 or tcp port 4500
```

Algoritmos, protocolos y políticas no coincidentes

Las conexiones VPN requieren que los puntos finales tengan algoritmos IKE, algoritmos IPsec y rangos de direcciones IP que coincidan. Si se produce un desajuste, la conexión falla. Si identifica un desajuste mediante uno de los siguientes métodos, arréglelo alineando algoritmos, protocolos o políticas.

- Si el extremo remoto no está ejecutando IKE/IPsec, puede ver un paquete ICMP indicándolo. Por ejemplo:

```
# ipsec auto --up vpn.example.com
...
000 "vpn.example.com"[1] 192.0.2.2 #16: ERROR: asynchronous network error report on
wlp2s0 (192.0.2.2:500), complainant 198.51.100.1: Connection refused [errno 111, origin
ICMP type 3 code 3 (not authenticated)]
...
```

- Ejemplo de algoritmos IKE no coincidentes:

```
# ipsec auto --up vpn.example.com
...
003 "vpn.example.com"[1] 193.110.157.148 #3: dropping unexpected IKE_SA_INIT message
containing NO_PROPOSAL_CHOSEN notification; message payloads: N; missing payloads:
SA,KE,Ni
```

- Ejemplo de algoritmos IPsec no coincidentes:

```
# ipsec auto --up vpn.example.com
...
182 "vpn.example.com"[1] 193.110.157.148 #5: STATE_PARENT_I2: sent v2I2, expected
v2R2 {auth=IKEv2 cipher=AES_GCM_16_256 integ=n/a prf=HMAC_SHA2_256
group=MODP2048}
002 "vpn.example.com"[1] 193.110.157.148 #6: IKE_AUTH response contained the error
notification NO_PROPOSAL_CHOSEN
```

  Una versión de IKE que no coincida también puede hacer que el punto final remoto abandone la solicitud sin respuesta. Esto parece idéntico a un cortafuegos que abandona todos los paquetes IKE.

- Ejemplo de rangos de direcciones IP no coincidentes para IKEv2 (llamados selectores de tráfico - TS):

```
# ipsec auto --up vpn.example.com
...
1v2 "vpn.example.com" #1: STATE_PARENT_I2: sent v2I2, expected v2R2 {auth=IKEv2
cipher=AES_GCM_16_256 integ=n/a prf=HMAC_SHA2_512 group=MODP2048}
002 "vpn.example.com" #2: IKE_AUTH response contained the error notification
TS_UNACCEPTABLE
```

- Ejemplo de rangos de direcciones IP no coincidentes para IKEv1:

```
# ipsec auto --up vpn.example.com
...
031 "vpn.example.com" #2: STATE_QUICK_I1: 60 second timeout exceeded after 0
retransmits.  No acceptable response to our first Quick Mode message: perhaps peer likes
no proposal
```

- Cuando se utilizan PreSharedKeys (PSK) en IKEv1, si ambas partes no ponen la misma PSK, todo el mensaje IKE se vuelve ilegible:

```
# ipsec auto --up vpn.example.com
```

```
...
003 "vpn.example.com" #1: received Hash Payload does not match computed value
223 "vpn.example.com" #1: sending notification INVALID_HASH_INFORMATION to
192.0.2.23:500
```

- En IKEv2, el error de PSK no coincidente resulta en un mensaje AUTHENTICATION_FAILED:

```
# ipsec auto --up vpn.example.com
...
002 "vpn.example.com" #1: IKE SA authentication request rejected by peer:
AUTHENTICATION_FAILED
```

Unidad máxima de transmisión

Aparte de los cortafuegos que bloquean los paquetes IKE o IPsec, la causa más común de los problemas de red está relacionada con el aumento del tamaño de los paquetes cifrados. El hardware de red fragmenta los paquetes más grandes que la unidad de transmisión máxima (MTU), por ejemplo, 1500 bytes. A menudo, los fragmentos se pierden y los paquetes no se vuelven a ensamblar. Esto provoca fallos intermitentes, cuando una prueba de ping, que utiliza paquetes de pequeño tamaño, funciona pero el resto del tráfico falla. En este caso, se puede establecer una sesión SSH pero el terminal se congela en cuanto se utiliza, por ejemplo, introduciendo el comando 'ls -al /usr' en el host remoto.

Para solucionar el problema, reduzca el tamaño de la MTU añadiendo la opción **mtu=1400** al archivo de configuración del túnel.

Alternativamente, para las conexiones TCP, active una regla iptables que cambie el valor de MSS:

```
# iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Si el comando anterior no resuelve el problema en su escenario, especifique directamente un tamaño menor en el parámetro **set-mss**:

```
# iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1380
```

Traducción de direcciones de red (NAT)

Cuando un host IPsec también sirve como router NAT, podría reasignar accidentalmente los paquetes. El siguiente ejemplo de configuración demuestra el problema:

```
conn myvpn
    left=172.16.0.1
    leftsubnet=10.0.2.0/24
    right=172.16.0.2
    rightsubnet=192.168.0.0/16
…
```

El sistema con dirección 172.16.0.1 tiene una regla NAT:

```
iptables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
```

Si el sistema en la dirección 10.0.2.33 envía un paquete a 192.168.0.1, el router traduce la fuente 10.0.2.33 a 172.16.0.1 antes de aplicar el cifrado IPsec.

Entonces, el paquete con la dirección de origen 10.0.2.33 ya no coincide con la configuración de **conn myvpn**, e IPsec no cifra este paquete.

Para resolver este problema, inserte reglas que excluyan NAT para los rangos de subred IPsec de destino en el router, en este ejemplo:

```
iptables -t nat -I POSTROUTING -s 10.0.2.0/24 -d 192.168.0.0/16 -j RETURN
```

Errores en el subsistema IPsec del kernel

El subsistema IPsec del kernel puede fallar, por ejemplo, cuando un error provoca una desincronización del espacio de usuario de IKE y del kernel IPsec. Para comprobar este tipo de problemas:

```
$ cat /proc/net/xfrm_stat
XfrmInError               0
XfrmInBufferError         0
...
```

Cualquier valor distinto de cero en la salida del comando anterior indica un problema. Si encuentra este problema, abra un nuevo caso de soporte, y adjunte la salida del comando anterior junto con los registros de IKE correspondientes.

Registros de Libreswan

Libreswan registra utilizando el protocolo **syslog** por defecto. Puedes utilizar el comando **journalctl** para encontrar entradas de registro relacionadas con IPsec. Como las entradas correspondientes al registro son enviadas por el demonio IKE de **pluto**, busque la palabra clave "pluto", por ejemplo:

```
$ journalctl -b | grep pluto
```

Para mostrar un registro en vivo para el servicio **ipsec**:

```
$ journalctl -f -u ipsec
```

Si el nivel de registro por defecto no revela su problema de configuración, active los registros de depuración añadiendo la opción **plutodebug=all** a la sección **config setup** del archivo **/etc/ipsec.conf**.

Tenga en cuenta que el registro de depuración produce muchas entradas, y es posible que el servicio **journald** o **syslogd** limite la velocidad de los mensajes **syslog**. Para asegurarse de que tiene registros completos, redirija el registro a un archivo. Edita el **/etc/ipsec.conf**, y añade el **logfile=/var/log/pluto.log** en la sección **config setup**.

Recursos adicionales

- Solución de problemas mediante archivos de registro
- Uso y configuración de firewalld
- **tcpdump(8)** y **ipsec.conf(5)** páginas man

3.12. INFORMACIÓN RELACIONADA

Los siguientes recursos proporcionan información adicional sobre **Libreswan** y el demonio **ipsec**.

Documentación instalada

- **ipsec(8)** página de manual - Describe las opciones de comando para **ipsec**.

- **ipsec.conf(5)** página de manual - Contiene información sobre la configuración de **ipsec**.
- **ipsec.secrets(5)** man page - Describe el formato del archivo **ipsec.secrets**.
- **ipsec_auto(8)** man page - Describe el uso del cliente de línea de comandos **auto** para manipular las conexiones IPsec de Libreswan establecidas mediante intercambios automáticos de claves.
- **ipsec_rsasigkey(8)** man page - Describe la herramienta utilizada para generar claves de firma RSA.
- **/usr/share/doc/libreswan-*version*/**

Documentación en línea

https://libreswan.org

El sitio web del proyecto de la corriente ascendente.

https://libreswan.org/wiki

La Wiki del Proyecto Libreswan.

https://libreswan.org/man/

Todas las páginas man de Libreswan.

Publicación especial 800-77 del NIST: Guía de VPNs IPsec

Orientación práctica a las organizaciones sobre la implantación de servicios de seguridad basados en IPsec.

CAPÍTULO 4. CONFIGURACIÓN DE MACSEC

La siguiente sección proporciona información sobre cómo configurar **Media Control Access Security** (**MACsec**), que es una tecnología de seguridad estándar 802.1AE IEEE para la comunicación segura en todo el tráfico de los enlaces Ethernet.

4.1. INTRODUCCIÓN A MACSEC

Media Access Control Security (**MACsec**, IEEE 802.1AE) cifra y autentifica todo el tráfico en las redes LAN con el algoritmo GCM-AES-128. **MACsec** puede proteger no sólo **IP** sino también el Protocolo de Resolución de Direcciones (ARP), el Descubrimiento de Vecinos (ND) o **DHCP**. Mientras que **IPsec** opera en la capa de red (capa 3) y **SSL** o **TLS** en la capa de aplicación (capa 7), **MACsec** opera en la capa de enlace de datos (capa 2). Combina **MACsec** con protocolos de seguridad para otras capas de red para aprovechar las diferentes características de seguridad que ofrecen estos estándares.

4.2. USO DE MACSEC CON LA HERRAMIENTA NMCLI

Este procedimiento muestra cómo configurar **MACsec** con la herramienta **nmcli**.

Requisitos previos

- El **NetworkManager** debe estar en funcionamiento.
- Ya tiene un CAK hexadecimal de 16 bytes (**$MKA_CAK**) y un CKN hexadecimal de 32 bytes (**$MKA_CKN**).

Procedimiento

1. Para añadir una nueva conexión utilizando **nmcli**, introduzca:

```
~]# nmcli connection add type macsec \
 con-name test-macsec+ ifname macsec0 \
 connection.autoconnect no \
 macsec.parent enp1s0 macsec.mode psk \
 macsec.mka-cak $MKA_CAK \
 macsec.mka-ckn $MKA_CKN
```

   Sustituya *macsec0* por el nombre del dispositivo que desea configurar.

2. Para activar la conexión, introduzca:

```
~]# nmcli connection up test-macsec
```

Después de este paso, el dispositivo *macsec0* está configurado y puede utilizarse para la conexión en red.

4.3. USO DE MACSEC CON WPA_SUPPLICANT

Este procedimiento muestra cómo habilitar **MACsec** con un conmutador que realiza la autenticación utilizando un par precompartido de clave de asociación de conectividad/nombre CAK (CAK/CKN).

Procedimiento

1. Crea un par CAK/CKN. Por ejemplo, el siguiente comando genera una clave de 16 bytes en notación hexadecimal:

```
~]$ dd if=/dev/urandom count=16 bs=1 2> /dev/null | hexdump -e '1/2 "%02x"'
```

2. Cree el archivo de configuración **wpa_supplicant.conf** y añada las siguientes líneas:

```
ctrl_interface=/var/run/wpa_supplicant
eapol_version=3
ap_scan=0
fast_reauth=1

network={
    key_mgmt=NONE
    eapol_flags=0
    macsec_policy=1

    mka_cak=0011... # 16 bytes hexadecimal
    mka_ckn=2233... # 32 bytes hexadecimal
}
```

   Utilice los valores del paso anterior para completar las líneas **mka_cak** y **mka_ckn** en el archivo de configuración **wpa_supplicant.conf**.

   Para más información, consulte la página de manual **wpa_supplicant.conf(5)**.

3. Asumiendo que está usando *wlp61s0* para conectarse a su red, inicie **wpa_supplicant** utilizando el siguiente comando:

```
~]# wpa_supplicant -i wlp61s0 -Dmacsec_linux -c wpa_supplicant.conf
```

4.4. INFORMACIÓN RELACIONADA

Para más detalles, consulte el artículo Novedades de MACsec: configuración de MACsec mediante wpa_supplicant y (opcionalmente) NetworkManager. Además, consulte el artículo MACsec: una solución diferente para cifrar el tráfico de red para obtener más información sobre la arquitectura de una red **MACsec**, escenarios de uso y ejemplos de configuración.

CAPÍTULO 5. USO Y CONFIGURACIÓN DE FIREWALLD

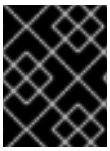
Un *firewall* es una forma de proteger las máquinas de cualquier tráfico no deseado procedente del exterior. Permite a los usuarios controlar el tráfico de red entrante en las máquinas anfitrionas definiendo un conjunto de *firewall rules*. Estas reglas se utilizan para clasificar el tráfico entrante y bloquearlo o permitirlo.

Tenga en cuenta que **firewalld** con el backend **nftables** no admite el paso de reglas personalizadas **nftables** a **firewalld**, utilizando la opción **--direct**.

5.1. CUÁNDO UTILIZAR FIREWALLD, NFTABLES O IPTABLES

A continuación se presenta un breve resumen en el que se debe utilizar una de las siguientes utilidades:

- **firewalld**: Utilice la utilidad **firewalld** para casos de uso de cortafuegos sencillos. La utilidad es fácil de usar y cubre los casos de uso típicos para estos escenarios.
- **nftables**: Utilice la utilidad **nftables** para configurar cortafuegos complejos y de rendimiento crítico, como por ejemplo para toda una red.
- **iptables**: La utilidad **iptables** en Red Hat Enterprise Linux 8 utiliza la API del kernel **nf_tables** en lugar del back end **legacy**. La API **nf_tables** proporciona compatibilidad con versiones anteriores para que los scripts que utilizan comandos **iptables** sigan funcionando en Red Hat Enterprise Linux 8. Para los nuevos scripts de cortafuegos, Red Hat recomienda utilizar **nftables**.



IMPORTANTE

Para evitar que los diferentes servicios de firewall se influyan mutuamente, ejecute sólo uno de ellos en un host RHEL y desactive los demás servicios.

5.2. CÓMO EMPEZAR CON FIREWALLD

5.2.1. firewalld

firewalld es un demonio de servicio de cortafuegos que proporciona un cortafuegos dinámico personalizable basado en el host con una interfaz **D-Bus**. Al ser dinámico, permite crear, cambiar y eliminar las reglas sin necesidad de reiniciar el demonio del cortafuegos cada vez que se cambian las reglas.

firewalld utiliza los conceptos de *zones* y *services*, que simplifican la gestión del tráfico. Las zonas son conjuntos predefinidos de reglas. Se pueden asignar interfaces de red y fuentes a una zona. El tráfico permitido depende de la red a la que esté conectado el ordenador y del nivel de seguridad que tenga asignado esta red. Los servicios del cortafuegos son reglas predefinidas que cubren todos los ajustes necesarios para permitir el tráfico entrante para un servicio específico y se aplican dentro de una zona.

Los servicios utilizan uno o más *ports* o *addresses* para la comunicación en red. Los cortafuegos filtran la comunicación basándose en los puertos. Para permitir el tráfico de red para un servicio, sus puertos deben ser *open*. **firewalld** bloquea todo el tráfico en los puertos que no están explícitamente establecidos como abiertos. Algunas zonas, como *trusted*, permiten todo el tráfico por defecto.

Recursos adicionales

- **firewalld(1)** página de manual

5.2.2. Zonas

firewalld puede utilizarse para separar las redes en diferentes zonas según el nivel de confianza que el usuario haya decidido otorgar a las interfaces y al tráfico dentro de esa red. Una conexión sólo puede formar parte de una zona, pero una zona puede utilizarse para muchas conexiones de red.

NetworkManager notifica a **firewalld** la zona de una interfaz. Puede asignar zonas a las interfaces con:

- **NetworkManager**
- **firewall-config** herramienta
- **firewall-cmd** herramienta de línea de comandos
- La consola web de RHEL

Los tres últimos sólo pueden editar los archivos de configuración correspondientes de **NetworkManager**. Si se cambia la zona de la interfaz mediante la consola web, **firewall-cmd** o **firewall-config**, la solicitud se reenvía a **NetworkManager** y no es gestionada por **firewalld**.

Las zonas predefinidas se almacenan en el directorio **/usr/lib/firewalld/zones/** y pueden aplicarse instantáneamente a cualquier interfaz de red disponible. Estos archivos se copian en el directorio **/etc/firewalld/zones/** sólo después de ser modificados. La configuración por defecto de las zonas predefinidas es la siguiente:

block

Cualquier conexión de red entrante es rechazada con un mensaje icmp-host-prohibido para **IPv4** e icmp6-adm-prohibido para **IPv6**. Sólo son posibles las conexiones de red iniciadas desde dentro del sistema.

dmz

Para los ordenadores de su zona desmilitarizada de acceso público con acceso limitado a su red interna. Sólo se aceptan las conexiones entrantes seleccionadas.

drop

Todos los paquetes de red entrantes se descartan sin ninguna notificación. Sólo son posibles las conexiones de red salientes.

external

Para usar en redes externas con el enmascaramiento activado, especialmente para los routers. No confía en que los otros ordenadores de la red no dañen su ordenador. Sólo se aceptan las conexiones entrantes seleccionadas.

home

Para usar en casa cuando se confía principalmente en los otros ordenadores de la red. Sólo se aceptan las conexiones entrantes seleccionadas.

internal

Para su uso en redes internas cuando se confía principalmente en los otros ordenadores de la red. Sólo se aceptan las conexiones entrantes seleccionadas.

public

Para su uso en áreas públicas donde no se confía en otros ordenadores de la red. Sólo se aceptan las conexiones entrantes seleccionadas.

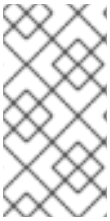
trusted

Se aceptan todas las conexiones de red.

work

Para su uso en el trabajo, donde se confía principalmente en los otros ordenadores de la red. Sólo se aceptan las conexiones entrantes seleccionadas.

Una de estas zonas se establece como la zona *default*. Cuando se añaden conexiones de interfaz a **NetworkManager**, se asignan a la zona por defecto. En la instalación, la zona por defecto en **firewalld** se establece como la zona **public**. La zona por defecto se puede cambiar.



NOTA

Los nombres de las zonas de red deben ser autoexplicativos y permitir a los usuarios tomar rápidamente una decisión razonable. Para evitar cualquier problema de seguridad, revise la configuración de la zona por defecto y desactive cualquier servicio innecesario según sus necesidades y evaluaciones de riesgo.

Recursos adicionales

- **firewalld.zone(5)** página de manual

5.2.3. Servicios predefinidos

Un servicio puede ser una lista de puertos locales, protocolos, puertos de origen y destinos, así como una lista de módulos de ayuda del cortafuegos que se cargan automáticamente si el servicio está activado. El uso de servicios ahorra tiempo a los usuarios porque pueden realizar varias tareas, como abrir puertos, definir protocolos, habilitar el reenvío de paquetes, etc., en un solo paso, en lugar de configurar todo uno tras otro.

Las opciones de configuración de los servicios y la información genérica de los archivos se describen en la página man **firewalld.service(5)**. Los servicios se especifican mediante archivos de configuración XML individuales, que se denominan con el siguiente formato ***service-name*.xml**. Se prefieren los nombres de los protocolos a los de los servicios o aplicaciones en **firewalld**.

Los servicios pueden añadirse y eliminarse mediante la herramienta gráfica **firewall-config**, **firewall-cmd**, y **firewall-offline-cmd**.

También puede editar los archivos XML en el directorio **/etc/firewalld/services/**. Si el usuario no añade o modifica un servicio, no se encuentra el archivo XML correspondiente en **/etc/firewalld/services/**. Los archivos del directorio **/usr/lib/firewalld/services/** pueden utilizarse como plantillas si se desea añadir o modificar un servicio.

Recursos adicionales

- **firewalld.service(5)** página de manual

5.3. INSTALACIÓN DE LA HERRAMIENTA DE CONFIGURACIÓN GUI FIREWALL-CONFIG

Para utilizar la herramienta de configuración GUI **firewall-config**, instale el paquete **firewall-config**.

Procedimiento

1. Introduzca el siguiente comando como **root**:

```
# yum install firewall-config
```

Alternativamente, en **GNOME, use the Super key and type `Software** para iniciar la aplicación **Software Sources**. Escriba **firewall** en el cuadro de búsqueda, que aparece después de seleccionar el botón de búsqueda en la esquina superior derecha. Seleccione el elemento **Firewall** de los resultados de la búsqueda y haga clic en el botón **Instalar**.

2. Para ejecutar **firewall-config**, utilice el comando **firewall-config** o pulse la tecla **Super** para entrar en **Activities Overview**, escriba **firewall** y pulse **Enter**.

5.4. VER EL ESTADO ACTUAL Y LA CONFIGURACIÓN DE FIREWALLD

5.4.1. Ver el estado actual de firewalld

El servicio de cortafuegos, **firewalld**, está instalado en el sistema por defecto. Utilice la interfaz CLI de **firewalld** para comprobar que el servicio se está ejecutando.

Procedimiento

1. Para ver el estado del servicio:

```
# firewall-cmd --state
```

2. Para obtener más información sobre el estado del servicio, utilice el subcomando **systemctl status**:

```
# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor pr
   Active: active (running) since Mon 2017-12-18 16:05:15 CET; 50min ago
     Docs: man:firewalld(1)
 Main PID: 705 (firewalld)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/firewalld.service
           └─705 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --nopid
```

Recursos adicionales

Es importante saber cómo está configurado **firewalld** y qué reglas están en vigor antes de intentar editar la configuración. Para mostrar la configuración del cortafuegos, consulte Sección 5.4.2, “Ver la configuración actual de firewalld”

5.4.2. Ver la configuración actual de firewalld

5.4.2.1. Visualización de los servicios permitidos mediante la GUI

Para ver la lista de servicios mediante la herramienta gráfica **firewall-config** pulse la tecla **Super** para acceder a la vista general de actividades, escriba **firewall** y pulse **Intro**. Aparece la herramienta **firewall-config** aparece la herramienta. Ahora puede ver la lista de servicios en la pestaña **Services**.

Como alternativa, para iniciar la herramienta gráfica de configuración del cortafuegos mediante la línea de comandos, introduzca el siguiente comando:

```
$ firewall-config
```

Se abre la ventana **Firewall Configuration**. Tenga en cuenta que este comando se puede ejecutar como un usuario normal, pero ocasionalmente se le pedirá una contraseña de administrador.

5.4.2.2. Visualización de la configuración de firewalld mediante la CLI

Con el cliente CLI, es posible obtener diferentes vistas de la configuración actual del cortafuegos. La opción **--list-all** muestra una visión completa de la configuración de **firewalld**.

firewalld utiliza zonas para gestionar el tráfico. Si no se especifica una zona mediante la opción **--zone**, el comando es efectivo en la zona por defecto asignada a la interfaz de red y la conexión activas.

Para listar toda la información relevante para la zona por defecto:

```
# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Para especificar la zona para la que se muestran los ajustes, añada el argumento **--zone=*zone-name*** al comando **firewall-cmd --list-all**, por ejemplo:

```
# firewall-cmd --list-all --zone=home
home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh mdns samba-client dhcpv6-client
... [trimmed for clarity]
```

Para ver la configuración de una información concreta, como los servicios o los puertos, utilice una opción específica. Consulte las páginas del manual **firewalld** u obtenga una lista de las opciones utilizando la ayuda del comando:

```
# firewall-cmd --help

Usage: firewall-cmd [OPTIONS...]

General Options
  -h, --help          Prints a short help text and exists
  -V, --version       Print the version string of firewalld
  -q, --quiet         Do not print status messages

Status Options
```

```
  --state             Return and print firewalld state
  --reload            Reload firewall and keep state information
... [trimmed for clarity]
```

Por ejemplo, para ver qué servicios están permitidos en la zona actual:

```
# firewall-cmd --list-services
ssh dhcpv6-client
```

NOTA

El listado de las configuraciones para una determinada subparte utilizando la herramienta CLI puede ser a veces difícil de interpretar. Por ejemplo, usted permite el servicio **SSH** y **firewalld** abre el puerto necesario (22) para el servicio. Más tarde, si se listan los servicios permitidos, la lista muestra el servicio **SSH**, pero si se listan los puertos abiertos, no muestra ninguno. Por lo tanto, se recomienda utilizar la opción **--list-all** para asegurarse de recibir una información completa.

5.5. INICIANDO FIREWALLD

Procedimiento

1. Para iniciar **firewalld**, introduzca el siguiente comando como **root**:

```
# systemctl unmask firewalld
# systemctl start firewalld
```

2. Para garantizar que **firewalld** se inicie automáticamente al arrancar el sistema, introduzca el siguiente comando como **root**:

```
# systemctl enable firewalld
```

5.6. DETENCIÓN DE FIREWALLD

Procedimiento

1. Para detener **firewalld**, introduzca el siguiente comando como **root**:

```
# systemctl stop firewalld
```

2. Para evitar que **firewalld** se inicie automáticamente al arrancar el sistema:

```
# systemctl disable firewalld
```

3. Para asegurarse de que firewalld no se inicia accediendo a la interfaz **firewalld D-Bus** y también si otros servicios requieren **firewalld**:

```
# systemctl mask firewalld
```

5.7. TIEMPO DE EJECUCIÓN Y AJUSTES PERMANENTES

Cualquier cambio realizado en el modo *runtime* sólo se aplica mientras **firewalld** está en funcionamiento. Cuando se reinicia **firewalld**, los ajustes vuelven a sus valores de *permanent*.

Para que los cambios persistan en los reinicios, aplíquelos de nuevo utilizando la opción **--permanent**. Alternativamente, para hacer que los cambios sean persistentes mientras se ejecuta **firewalld**, utilice la opción **--runtime-to-permanent firewall-cmd** .

Si establece las reglas mientras **firewalld** se está ejecutando utilizando sólo la opción **--permanent**, no se hacen efectivas antes de que se reinicie **firewalld**. Sin embargo, al reiniciar **firewalld** se cierran todos los puertos abiertos y se detiene el tráfico de red.

Modificación de ajustes en tiempo de ejecución y configuración permanente mediante CLI

Utilizando la CLI, no se modifica la configuración del cortafuegos en ambos modos al mismo tiempo. Sólo se modifica el modo de tiempo de ejecución o el modo permanente. Para modificar la configuración del cortafuegos en el modo permanente, utilice la opción **--permanent** con el comando **firewall-cmd**.

```
# firewall-cmd --permanent <other options>
```

Sin esta opción, el comando modifica el modo de ejecución.

Para cambiar la configuración en ambos modos, puedes utilizar dos métodos:

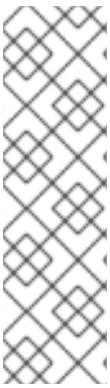
1. Cambie la configuración de tiempo de ejecución y luego hágala permanente de la siguiente manera:

```
# firewall-cmd <other options>
# firewall-cmd --runtime-to-permanent
```

2. Establezca los ajustes permanentes y recargue los ajustes en el modo de ejecución:

```
# firewall-cmd --permanent <other options>
# firewall-cmd --reload
```

El primer método permite probar los ajustes antes de aplicarlos al modo permanente.



NOTA

Es posible, especialmente en los sistemas remotos, que una configuración incorrecta haga que un usuario se bloquee en una máquina. Para evitar estas situaciones, utilice la opción **--timeout**. Después de un tiempo determinado, cualquier cambio vuelve a su estado anterior. El uso de esta opción excluye la opción **--permanent**.

Por ejemplo, para añadir el servicio **SSH** durante 15 minutos:

```
# firewall-cmd --add-service=ssh --timeout 15m
```

5.8. VERIFICACIÓN DE LA CONFIGURACIÓN PERMANENTE DE FIREWALLD

En ciertas situaciones, por ejemplo después de editar manualmente los archivos de configuración de **firewalld**, los administradores quieren verificar que los cambios son correctos. Esta sección describe cómo verificar la configuración permanente del servicio **firewalld**.

Requisitos previos

- El servicio **firewalld** está funcionando.

Procedimiento

1. Verifique la configuración permanente del servicio **firewalld**:

```
# firewall-cmd --check-config
success
```

Si la configuración permanente es válida, el comando devuelve **success**. En otros casos, el comando devuelve un error con más detalles, como el siguiente:

```
# firewall-cmd --check-config
Error: INVALID_PROTOCOL: 'public.xml': 'tcpx' not from {'tcp'|'udp'|'sctp'|'dccp'}
```

5.9. CONTROLAR EL TRÁFICO DE LA RED MEDIANTE FIREWALLD

5.9.1. Desactivación de todo el tráfico en caso de emergencia mediante CLI

En una situación de emergencia, como un ataque al sistema, es posible desactivar todo el tráfico de la red y cortar al atacante.

Procedimiento

1. Para desactivar inmediatamente el tráfico de red, active el modo de pánico:

```
# firewall-cmd --panic-on
```

IMPORTANTE

La activación del modo pánico detiene todo el tráfico de red. Por esta razón, debe ser utilizado sólo cuando se tiene el acceso físico a la máquina o si se inicia la sesión utilizando una consola serie.

Al desactivar el modo de pánico, el cortafuegos vuelve a su configuración permanente. Para desactivar el modo de pánico:

```
# firewall-cmd --panic-off
```

Para ver si el modo de pánico está activado o desactivado, utilice:

```
# firewall-cmd --query-panic
```

5.9.2. Control del tráfico con servicios predefinidos mediante CLI

El método más sencillo para controlar el tráfico es añadir un servicio predefinido a **firewalld**. Así se abren todos los puertos necesarios y se modifican otras configuraciones según el *service definition file* .

Procedimiento

1. Compruebe que el servicio no está ya permitido:

```
# firewall-cmd --list-services
ssh dhcpv6-client
```

2. Enumerar todos los servicios predefinidos:

```
# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bitcoin bitcoin-rpc
bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6
dhcpv6-client dns docker-registry ...
[trimmed for clarity]
```

3. Añade el servicio a los servicios permitidos:

```
# firewall-cmd --add-service=<nombre-servicio>
```

4. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5.9.3. Control del tráfico con servicios predefinidos mediante la interfaz gráfica de usuario

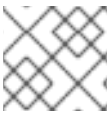
Para activar o desactivar un servicio predefinido o personalizado:

1. Inicie la herramienta **firewall-config** herramienta y seleccione la zona de red cuyos servicios se van a configurar.
2. Seleccione la pestaña **Services**.
3. Seleccione la casilla de cada tipo de servicio en el que desee confiar o desactive la casilla para bloquear un servicio.

Para editar un servicio:

1. Inicie la **firewall-config** herramienta.
2. Seleccione **Permanent** en el menú denominado **Configuration**. En la parte inferior de la ventana de **Servicios** aparecen otros iconos y botones de menú.
3. Seleccione el servicio que desea configurar.

Las pestañas **Ports**, **Protocols**, y **Source Port** permiten añadir, cambiar y eliminar puertos, protocolos y puerto de origen para el servicio seleccionado. La pestaña de módulos es para configurar **Netfilter** módulos de ayuda. La pestaña **Destination** permite limitar el tráfico a una dirección de destino y un protocolo de Internet concretos (**IPv4** o **IPv6**).

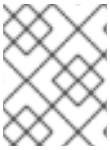


NOTA

No es posible modificar los ajustes de servicio en el modo **Runtime**.

5.9.4. Añadir nuevos servicios

Los servicios pueden añadirse y eliminarse utilizando la herramienta gráfica **firewall-config** **firewall-cmd** y **firewall-offline-cmd**. También se pueden editar los archivos XML en **/etc/firewalld/services/**. Si el usuario no añade o modifica un servicio, no se encuentra el archivo XML correspondiente en **/etc/firewalld/services/**. Los archivos **/usr/lib/firewalld/services/** pueden utilizarse como plantillas si se desea añadir o modificar un servicio.



NOTA

Los nombres de los servicios deben ser alfanuméricos y, además, sólo pueden incluir los caracteres _ (guión bajo) y **-** (guión).

Procedimiento

Para añadir un nuevo servicio en un terminal, utilice **firewall-cmd**, o **firewall-offline-cmd** en caso de que no esté activo **firewalld**.

1. Introduzca el siguiente comando para añadir un servicio nuevo y vacío:

```
$ firewall-cmd --new-service=service-name --permanent
```

2. Para añadir un nuevo servicio utilizando un archivo local, utilice el siguiente comando:

```
$ firewall-cmd --new-service-from-file=service-name.xml --permanent
```

   Puede cambiar el nombre del servicio con la opción adicional **--name=*service-name*** adicional.

3. En cuanto se modifican los ajustes del servicio, se coloca una copia actualizada del mismo en **/etc/firewalld/services/**.
   Como **root**, puede introducir el siguiente comando para copiar un servicio manualmente:

```
# cp /usr/lib/firewalld/services/service-name.xml /etc/firewalld/services/service-name.xml
```

firewalld carga los archivos de **/usr/lib/firewalld/services** en primer lugar. Si los archivos se colocan en **/etc/firewalld/services** y son válidos, entonces estos anularán los archivos coincidentes de **/usr/lib/firewalld/services**. Los archivos anulados en **/usr/lib/firewalld/services** se utilizan tan pronto como los archivos coincidentes en **/etc/firewalld/services** se hayan eliminado o si se ha pedido a **firewalld** que cargue los valores predeterminados de los servicios. Esto se aplica sólo al entorno permanente. Se necesita una recarga para obtener estos fallbacks también en el entorno de ejecución.

5.9.5. Controlar los puertos mediante la CLI

Los puertos son dispositivos lógicos que permiten a un sistema operativo recibir y distinguir el tráfico de red y reenviarlo en consecuencia a los servicios del sistema. Suelen estar representados por un demonio que escucha en el puerto, es decir, que espera cualquier tráfico que llegue a este puerto.

Normalmente, los servicios del sistema escuchan en los puertos estándar que están reservados para ellos. El demonio **httpd**, por ejemplo, escucha en el puerto 80. Sin embargo, los administradores del sistema configuran por defecto los dæmons para que escuchen en diferentes puertos para mejorar la seguridad o por otras razones.

5.9.5.1. Abrir un puerto

A través de los puertos abiertos, el sistema es accesible desde el exterior, lo que representa un riesgo de seguridad. Por lo general, mantén los puertos cerrados y sólo ábrelos si son necesarios para determinados servicios.

Procedimiento

Para obtener una lista de puertos abiertos en la zona actual:

1. Lista de todos los puertos permitidos:

```
# firewall-cmd --list-ports
```

2. Añade un puerto a los puertos permitidos para abrirlo al tráfico entrante:

```
# firewall-cmd --add-port=número de puerto/tipo de puerto
```

3. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

Los tipos de puerto son **tcp**, **udp**, **sctp**, o **dccp**. El tipo debe coincidir con el tipo de comunicación de red.

5.9.5.2. Cerrar un puerto

Cuando un puerto abierto ya no es necesario, cierre ese puerto en **firewalld**. Se recomienda encarecidamente cerrar todos los puertos innecesarios en cuanto no se utilicen, ya que dejar un puerto abierto representa un riesgo para la seguridad.

Procedimiento

Para cerrar un puerto, elimínelo de la lista de puertos permitidos:

1. Lista de todos los puertos permitidos:

```
# firewall-cmd --list-ports
[WARNING]
====
This command will only give you a list of ports that have been opened as ports. You will not be able to see any open ports that have been opened as a service. Therefore, you should consider using the --list-all option instead of --list-ports.
====
```

2. Elimina el puerto de los puertos permitidos para cerrarlo al tráfico entrante:

```
# firewall-cmd --remove-port=número de puerto/tipo de puerto
```

3. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5.9.6. Abrir puertos mediante la GUI

Para permitir el tráfico a través del firewall a un determinado puerto:

1. Inicie la **firewall-config** herramienta y seleccione la zona de red cuya configuración desea modificar.

2. Seleccione la pestaña **Ports** y haga clic en el botón **Añadir** de la derecha. Se abre la ventana **Port and Protocol**.

3. Introduzca el número de puerto o el rango de puertos a permitir.

4. Seleccione **tcp** o **udp** de la lista.

5.9.7. Control del tráfico con protocolos mediante GUI

Para permitir el tráfico a través del cortafuegos utilizando un determinado protocolo:

1. Inicie la **firewall-config** herramienta y seleccione la zona de red cuya configuración desea modificar.

2. Seleccione la pestaña **Protocols** y haga clic en el botón **Add** de la derecha. Se abre la ventana **Protocol**.

3. Seleccione un protocolo de la lista o marque la casilla **Other Protocol** e introduzca el protocolo en el campo.

5.9.8. Abrir puertos de origen mediante la GUI

Para permitir el tráfico a través del cortafuegos desde un determinado puerto:

1. Inicie la herramienta de configuración del cortafuegos y seleccione la zona de red cuya configuración desea modificar.

2. Seleccione la pestaña **Source Port** y haga clic en el botón **Add** de la derecha. Se abre la ventana **Source Port**.

3. Introduzca el número de puerto o el rango de puertos a permitir. Seleccione **tcp** o **udp** de la lista.

5.10. TRABAJAR CON ZONAS DE FIREWALLD

Las zonas representan un concepto para gestionar el tráfico entrante de forma más transparente. Las zonas están conectadas a interfaces de red o se les asigna un rango de direcciones de origen. Las reglas del cortafuegos se gestionan para cada zona de forma independiente, lo que permite definir configuraciones complejas del cortafuegos y aplicarlas al tráfico.

5.10.1. Zonas de cotización

Procedimiento

1. Para ver qué zonas están disponibles en su sistema:

```
# firewall-cmd --get-zones
```

   El comando **firewall-cmd --get-zones** muestra todas las zonas que están disponibles en el sistema, pero no muestra ningún detalle de zonas concretas.

2. Para ver información detallada de todas las zonas:

```
# firewall-cmd --list-all-zones
```

3. Para ver información detallada de una zona concreta:

```
# firewall-cmd --zone=nombre-de-la-zona --list-all
```

5.10.2. Modificación de la configuración de firewalld para una zona determinada

En Sección 5.9.2, "Control del tráfico con servicios predefinidos mediante CLI" y Sección 5.9.5, "Controlar los puertos mediante la CLI" se explica cómo añadir servicios o modificar puertos en el ámbito de la zona de trabajo actual. A veces, es necesario establecer reglas en una zona diferente.

Procedimiento

1. Para trabajar en una zona diferente, utilice la opción **--zone=*zone-name*** opción. Por ejemplo, para permitir el servicio **SSH** en la zona *public*:

```
# firewall-cmd --add-service=ssh --zone=public
```

5.10.3. Cambiar la zona por defecto

Los administradores del sistema asignan una zona a una interfaz de red en sus archivos de configuración. Si una interfaz no está asignada a una zona específica, se asigna a la zona por defecto. Después de cada reinicio del servicio **firewalld**, **firewalld** carga la configuración de la zona por defecto y la activa.

Procedimiento

Para configurar la zona por defecto:

1. Muestra la zona actual por defecto:

```
# firewall-cmd --get-default-zone
```

2. Establezca la nueva zona por defecto:

```
# firewall-cmd --set-default-zone zone-name
```

NOTA

Siguiendo este procedimiento, el ajuste es permanente, incluso sin la opción **--permanent**.

5.10.4. Asignación de una interfaz de red a una zona

Es posible definir diferentes conjuntos de reglas para diferentes zonas y luego cambiar la configuración rápidamente cambiando la zona para la interfaz que se está utilizando. Con múltiples interfaces, se puede establecer una zona específica para cada una de ellas para distinguir el tráfico que llega a través de ellas.

Procedimiento

Para asignar la zona a una interfaz específica:

1. Enumera las zonas activas y las interfaces asignadas a ellas:

```
# firewall-cmd --get-active-zones
```

2. Asigne la interfaz a una zona diferente:

```
# firewall-cmd --zone=zone_name --change-interface=interface_name --permanent
```

5.10.5. Asignación de una zona a una conexión mediante nmcli

Este procedimiento describe cómo añadir una zona de firewalld a una conexión de NetworkManager utilizando la utilidad **nmcli**.

Procedimiento

1. Asigna la zona al perfil de conexión de NetworkManager:

```
# nmcli connection modify profile connection.zone zone_name
```

2. Recarga la conexión:

```
# nmcli connection up profile
```

5.10.6. Asignación manual de una zona a una conexión de red en un archivo ifcfg

Cuando la conexión es gestionada por **NetworkManager**debe conocer una zona que utiliza. Para cada conexión de red, se puede especificar una zona, lo que proporciona la flexibilidad de varias configuraciones de cortafuegos según la ubicación del ordenador con dispositivos portátiles. Así, se pueden especificar zonas y configuraciones para diferentes ubicaciones, como la empresa o el hogar.

Procedimiento

1. Para establecer una zona para una conexión, edite el archivo **/etc/sysconfig/network-scripts/ifcfg-*connection_name*** y añada una línea que asigne una zona a esta conexión:

```
ZONA=zone_name
```

5.10.7. Crear una nueva zona

Para utilizar zonas personalizadas, cree una nueva zona y utilícela igual que una zona predefinida. Las nuevas zonas requieren la opción **--permanent**, de lo contrario el comando no funciona.

Procedimiento

Para crear una nueva zona:

1. Crear una nueva zona:

```
# firewall-cmd --new-zone=nombre-de-zona
```

2. Compruebe si la nueva zona se ha añadido a su configuración permanente:

```
# firewall-cmd --get-zones
```

3. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5.10.8. Archivos de configuración de zona

Las zonas también pueden crearse utilizando un *zone configuration file*. Este enfoque puede ser útil cuando se necesita crear una nueva zona, pero se quiere reutilizar la configuración de otra zona y sólo alterarla un poco.

Un archivo de configuración de zona **firewalld** contiene la información de una zona. Se trata de la descripción de la zona, los servicios, los puertos, los protocolos, los bloqueos icmp, la mascarada, los puertos de reenvío y las reglas de lenguaje enriquecido en formato de archivo XML. El nombre del archivo debe ser ***zone-name*.xml** donde la longitud de *zone-name* está limitada actualmente a 17 caracteres. Los archivos de configuración de zona se encuentran en los directorios **/usr/lib/firewalld/zones/** y **/etc/firewalld/zones/**.

El siguiente ejemplo muestra una configuración que permite un servicio (**SSH**) y un rango de puertos, tanto para los protocolos **TCP** como **UDP**:

```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>My zone</short>
  <description>Here you can describe the characteristic features of the zone.</description>
  <service name="ssh"/>
  <port port="1025-65535" protocol="tcp"/>
  <port port="1025-65535" protocol="udp"/>
</zone>
```

Para cambiar la configuración de esa zona, añada o elimine secciones para añadir puertos, reenviar puertos, servicios, etc.

Recursos adicionales

- Para más información, consulte las páginas del manual **firewalld.zone**.

5.10.9. Uso de objetivos de zona para establecer el comportamiento por defecto para el tráfico entrante

Para cada zona, se puede establecer un comportamiento por defecto que gestione el tráfico entrante que no se especifique más. Este comportamiento se define estableciendo el objetivo de la zona. Hay cuatro opciones - **default**, **ACCEPT**, **REJECT**, y **DROP**. Si se establece el objetivo en **ACCEPT**, se aceptan todos los paquetes entrantes excepto los deshabilitados por una regla específica. Si establece el objetivo en **REJECT** o **DROP**, deshabilita todos los paquetes entrantes excepto los que haya permitido en reglas específicas. Cuando los paquetes son rechazados, la máquina de origen es informada del rechazo, mientras que no se envía ninguna información cuando los paquetes son descartados.

Procedimiento

Para establecer un objetivo para una zona:

1. Enumerar la información de la zona específica para ver el objetivo por defecto:

```
$ firewall-cmd --zone=zone-name --list-all
```

2. Establece un nuevo objetivo en la zona:

```
# firewall-cmd --permanent --zone=nombre-de-zona --set-target=
<default|ACCEPT|REJECT|DROP>
```

5.11. USO DE ZONAS PARA GESTIONAR EL TRÁFICO ENTRANTE EN FUNCIÓN DE UNA FUENTE

5.11.1. Uso de zonas para gestionar el tráfico entrante en función de una fuente

Puede utilizar las zonas para gestionar el tráfico entrante en función de su origen. Esto le permite clasificar el tráfico entrante y enrutarlo a través de diferentes zonas para permitir o rechazar los servicios que pueden ser alcanzados por ese tráfico.

Si añade una fuente a una zona, ésta se activa y todo el tráfico entrante procedente de esa fuente será dirigido a través de ella. Puede especificar diferentes configuraciones para cada zona, que se aplican al tráfico de las fuentes dadas en consecuencia. Puede utilizar más zonas aunque sólo tenga una interfaz de red.

5.11.2. Añadir una fuente

Para enrutar el tráfico entrante hacia una fuente específica, añada la fuente a esa zona. El origen puede ser una dirección IP o una máscara IP en la notación Classless Inter-domain Routing (CIDR).

NOTA

En caso de que añada varias zonas con un rango de red superpuesto, se ordenan alfanuméricamente por nombre de zona y sólo se tiene en cuenta la primera.

- Para fijar la fuente en la zona actual:

```
# firewall-cmd --add-source=<source>
```

- Para establecer la dirección IP de origen para una zona específica:

```
# firewall-cmd --zone=nombre-de-la-zona --add-source=<source>
```

El siguiente procedimiento permite todo el tráfico entrante de *192.168.2.15* en la zona **trusted**:

Procedimiento

1. Enumera todas las zonas disponibles:

```
# firewall-cmd --get-zones
```

2. Añade la IP de origen a la zona de confianza en el modo permanente:

```
# firewall-cmd --zone=trusted --add-source=192.168.2.15
```

3. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5.11.3. Eliminar una fuente

Al eliminar una fuente de la zona se corta el tráfico procedente de ella.

Procedimiento

1. Lista de fuentes permitidas para la zona requerida:

```
# firewall-cmd --zone=nombre-de-la-zona --list-sources
```

2. Eliminar la fuente de la zona de forma permanente:

```
# firewall-cmd --zone=nombre-de-zona --remove-source=<source>
```

3. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5.11.4. Añadir un puerto de origen

Para habilitar la clasificación del tráfico basada en un puerto de origen, especifique un puerto de origen utilizando la opción **--add-source-port**. También puede combinar esto con la opción **--add-source** para limitar el tráfico a una determinada dirección IP o rango de IP.

Procedimiento

1. Para añadir un puerto de origen:

```
# firewall-cmd --zone=nombre-de-la-zona --add-source-port=-nombre-de-
puerto>/<tcp|udp|sctp|dccp>
```

5.11.5. Eliminación de un puerto de origen

Al eliminar un puerto de origen se desactiva la clasificación del tráfico en función de un puerto de origen.

Procedimiento

1. Para eliminar un puerto de origen:

```
# firewall-cmd --zone=nombre-de-la-zona --remove-source-port=-nombre-de-
puerto>/<tcp|udp|sctp|dccp>
```

5.11.6. Uso de zonas y fuentes para permitir un servicio sólo para un dominio específico

Para permitir que el tráfico de una red específica utilice un servicio en una máquina, utilice zonas y origen. El siguiente procedimiento permite que el tráfico de *192.168.1.0/24* pueda llegar al servicio *HTTP* mientras que cualquier otro tráfico está bloqueado.

Procedimiento

1. Enumera todas las zonas disponibles:

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

2. Añade el origen a la zona de confianza para enrutar el tráfico originado por el origen a través de la zona:

```
# firewall-cmd --zone=trusted --add-source=192.168.1.0/24
```

3. Añade el servicio *http* en la zona de confianza:

```
# firewall-cmd --zone=trusted --add-service=http
```

4. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5. Compruebe que la zona de confianza está activa y que el servicio está permitido en ella:

```
# firewall-cmd --zone=trusted --list-all
trusted (active)
target: ACCEPT
sources: 192.168.1.0/24
services: http
```

5.11.7. Configurar el tráfico aceptado por una zona en función de un protocolo

Puede permitir que el tráfico entrante sea aceptado por una zona basándose en un protocolo. Todo el tráfico que utiliza el protocolo especificado es aceptado por una zona, en la que puede aplicar más reglas y filtrado.

5.11.7.1. Añadir un protocolo a una zona

Al añadir un protocolo a una zona determinada, se permite que todo el tráfico con este protocolo sea aceptado por esta zona.

Procedimiento

1. Para añadir un protocolo a una zona:

```
# firewall-cmd --zone=nombre-de-zona --add-protocolo=nombre-de-
puerto/tcp|udp|sctp|dccp|igmp
```

NOTA

Para recibir tráfico de multidifusión, utilice el valor **igmp** con la opción **--add-protocol**.

5.11.7.2. Eliminar un protocolo de una zona

Al eliminar un protocolo de una zona determinada, se deja de aceptar todo el tráfico basado en este protocolo por la zona.

Procedimiento

1. Para eliminar un protocolo de una zona:

```
# firewall-cmd --zone=nombre-de-zona --remove-protocolo=nombre-de-puerto/tcp|udp|sctp|dccp|igmp
```

5.12. CONFIGURACIÓN DEL ENMASCARAMIENTO DE DIRECCIONES IP

El siguiente procedimiento describe cómo habilitar el enmascaramiento de IP en su sistema. El enmascaramiento de IP oculta las máquinas individuales detrás de una puerta de enlace cuando se accede a Internet.

Procedimiento

1. Para comprobar si el enmascaramiento de IP está activado (por ejemplo, para la zona **external**), introduzca el siguiente comando como **root**:

```
# firewall-cmd --zone=external --query-masquerade
```

   El comando imprime **yes** con el estado de salida **0** si está activado. Imprime **no** con el estado de salida **1** en caso contrario. Si se omite **zone**, se utilizará la zona por defecto.

2. Para activar el enmascaramiento de IP, introduzca el siguiente comando como **root**:

```
# firewall-cmd --zone=external --add-masquerade
```

3. Para que esta configuración sea persistente, repita el comando añadiendo la opción **--permanent**.

Para desactivar el enmascaramiento de IP, introduzca el siguiente comando como **root**:

```
# firewall-cmd --zone=external --remove-masquerade --permanent
```

5.13. REENVÍO DE PUERTOS

El redireccionamiento de puertos mediante este método sólo funciona para el tráfico basado en IPv4. Para la configuración de redireccionamiento de IPv6, debe utilizar reglas ricas.

Para redirigir a un sistema externo, es necesario habilitar el enmascaramiento. Para más información, consulte Configuración del enmascaramiento de direcciones IP.

5.13.1. Añadir un puerto para redirigir

Utilizando **firewalld**, puede configurar la redirección de puertos para que cualquier tráfico entrante que llegue a un determinado puerto de su sistema sea entregado a otro puerto interno de su elección o a un puerto externo de otra máquina.

Requisitos previos

- Antes de redirigir el tráfico de un puerto a otro puerto, o a otra dirección, hay que saber tres cosas: a qué puerto llegan los paquetes, qué protocolo se utiliza y a dónde se quiere redirigir.

Procedimiento

Para redirigir un puerto a otro puerto:

```
# firewall-cmd --add-forward-port=puerto=número de puerto:proto=tcp|udp|sctp|dccp:toport=número de puerto
```

Para redirigir un puerto a otro puerto en una dirección IP diferente:

1. Añade el puerto a reenviar:

```
# firewall-cmd --add-forward-port=puerto=número de puerto:proto=tcp|udp:toport=número de puerto:toaddr=IP
```

2. Habilitar la mascarada:

```
# firewall-cmd --add-masquerade
```

5.13.2. Redirigir el puerto TCP 80 al puerto 88 en la misma máquina

Siga los pasos para redirigir el puerto TCP 80 al puerto 88.

Procedimiento

1. Redirige el puerto 80 al puerto 88 para el tráfico TCP:

```
# firewall-cmd --add-forward-port=80:proto=tcp:toport=88
```

2. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

3. Compruebe que el puerto está redirigido:

```
# firewall-cmd --list-all
```

5.13.3. Eliminación de un puerto redirigido

Para eliminar un puerto redirigido:

```
# firewall-cmd --remove-forward-port=puerto=número de puerto:proto=<tcp|udp>:toport=número de puerto:toaddr=<IP>
```

Para eliminar un puerto redirigido a una dirección diferente, utilice el siguiente procedimiento.

Procedimiento

1. Elimina el puerto reenviado:

```
# firewall-cmd --remove-forward-port=puerto=número de puerto:proto=
<tcp|udp>:toport=número de puerto:toaddr=<IP>
```

2. Desactivar la mascarada:

```
# firewall-cmd --remove-masquerade
```

5.13.4. Eliminación del puerto TCP 80 reenviado al puerto 88 en la misma máquina

Para eliminar la redirección de puertos:

Procedimiento

1. Lista de puertos redirigidos:

```
~]# firewall-cmd --list-forward-ports
port=80:proto=tcp:toport=88:toaddr=
```

2. Elimine el puerto redirigido del cortafuegos::

```
~]# firewall-cmd  --remove-forward-port=port=80:proto=tcp:toport=88:toaddr=
```

3. Haz que la nueva configuración sea persistente:

```
~]# firewall-cmd --runtime-to-permanent
```

5.14. GESTIÓN DE PETICIONES ICMP

El **Internet Control Message Protocol** (**ICMP**) es un protocolo de apoyo que utilizan varios dispositivos de red para enviar mensajes de error e información operativa que indican un problema de conexión, por ejemplo, que un servicio solicitado no está disponible. **ICMP** se diferencia de los protocolos de transporte como TCP y UDP porque no se utiliza para intercambiar datos entre sistemas.

Lamentablemente, es posible utilizar los mensajes de **ICMP**, especialmente **echo-request** y **echo-reply**, para revelar información sobre su red y hacer un mal uso de dicha información para diversos tipos de actividades fraudulentas. Por lo tanto, **firewalld** permite bloquear las solicitudes de **ICMP** para proteger la información de su red.

5.14.1. Listado y bloqueo de peticiones ICMP

Listado ICMP solicitudes

Las solicitudes de **ICMP** se describen en archivos XML individuales que se encuentran en el directorio **/usr/lib/firewalld/icmptypes/**. Puede leer estos archivos para ver una descripción de la solicitud. El comando **firewall-cmd** controla la manipulación de las peticiones de **ICMP**.

- Para listar todos los tipos disponibles de **ICMP**:

```
# firewall-cmd --get-icmptypes
```

- La petición **ICMP** puede ser utilizada por IPv4, IPv6 o por ambos protocolos. Para ver para qué protocolo se utiliza la petición **ICMP**:

```
# firewall-cmd --info-icmptype=<icmptype>
```

- El estado de una solicitud **ICMP** muestra **yes** si la solicitud está actualmente bloqueada o **no** si no lo está. Para ver si una solicitud de **ICMP** está actualmente bloqueada:

```
# firewall-cmd --query-icmp-block=<icmptype>
```

Bloqueo o desbloqueo de las solicitudes de ICMP

Cuando su servidor bloquea las solicitudes de **ICMP**, no proporciona la información que normalmente proporcionaría. Sin embargo, eso no significa que no se proporcione ninguna información. Los clientes reciben información de que la petición **ICMP** en particular está siendo bloqueada (rechazada). El bloqueo de las peticiones a **ICMP** debe considerarse cuidadosamente, porque puede causar problemas de comunicación, especialmente con el tráfico IPv6.

- Para ver si una solicitud de **ICMP** está actualmente bloqueada:

```
# firewall-cmd --query-icmp-block=<icmptype>
```

- Para bloquear una solicitud de **ICMP**:

```
# firewall-cmd --add-icmp-block=<icmptype>
```

- Para eliminar el bloqueo de una solicitud de **ICMP**:

```
# firewall-cmd --remove-icmp-block=<icmptype>
```

Bloqueo de las solicitudes de ICMP sin proporcionar ninguna información

Normalmente, si bloqueas las peticiones de **ICMP**, los clientes saben que lo estás bloqueando. Por lo tanto, un atacante potencial que esté husmeando en busca de direcciones IP activas todavía es capaz de ver que tu dirección IP está en línea. Para ocultar esta información por completo, tienes que eliminar todas las peticiones de **ICMP**.

- Para bloquear y abandonar todas las solicitudes de **ICMP**:

1. Establezca el objetivo de su zona en **DROP**:

```
# firewall-cmd --permanent --set-target=DROP
```

Ahora, todo el tráfico, incluyendo las solicitudes de **ICMP**, se descarta, excepto el tráfico que usted ha permitido explícitamente.

- Para bloquear y eliminar ciertas solicitudes de **ICMP** y permitir otras:

1. Establezca el objetivo de su zona en **DROP**:

```
# firewall-cmd --permanent --set-target=DROP
```

2. Añade la inversión de bloque ICMP para bloquear todas las peticiones de **ICMP** a la vez:

```
# firewall-cmd --add-icmp-block-inversion
```

3. Añade el bloqueo ICMP para aquellas peticiones de **ICMP** que quieras permitir:

```
# firewall-cmd --add-icmp-block=<icmptype>
```

4. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

El *block inversion* invierte la configuración de los bloqueos de las peticiones de **ICMP**, por lo que todas las peticiones, que antes no estaban bloqueadas, se bloquean debido a que el objetivo de su zona cambia a **DROP**. Las peticiones que estaban bloqueadas no lo están. Esto significa que si quiere desbloquear una petición, debe utilizar el comando de bloqueo.

- Para revertir la inversión de bloques a una configuración totalmente permisiva:

1. Establezca el objetivo de su zona en **default** o **ACCEPT**:

```
# firewall-cmd --permanent --set-target=default
```

2. Eliminar todos los bloques añadidos para las solicitudes de **ICMP**:

```
# firewall-cmd --remove-icmp-block=<icmptype>
```

3. Retire la inversión del bloque **ICMP**:

```
# firewall-cmd --remove-icmp-block-inversion
```

4. Haz que la nueva configuración sea persistente:

```
# firewall-cmd --runtime-to-permanent
```

5.14.2. Configuración del filtro ICMP mediante la GUI

- Para activar o desactivar un filtro **ICMP**, inicie la herramienta **firewall-config** herramienta y seleccione la zona de red cuyos mensajes deben ser filtrados. Seleccione la pestaña **ICMP Filter** y marque la casilla correspondiente a cada tipo de mensaje **ICMP** que desee filtrar. Desactive la casilla para desactivar un filtro. Esta configuración es por dirección y el valor por defecto permite todo.

- Para editar un tipo de **ICMP**, inicie la herramienta **firewall-config** herramienta y seleccione el modo **Permanent** en el menú denominado **Configuration**. Aparecen iconos adicionales en la parte inferior de la ventana de **Servicios**. Seleccione **Sí** en el siguiente cuadro de diálogo para habilitar el enmascaramiento y hacer que funcione el reenvío a otra máquina.

- Para activar la inversión del **ICMP Filter**, haga clic en la casilla **Invert Filter** de la derecha. Ahora sólo se aceptan los tipos de **ICMP** marcados, todos los demás se rechazan. En una zona que utilice el objetivo DROP, se descartan.

5.15. AJUSTE Y CONTROL DE LOS CONJUNTOS IP MEDIANTE FIREWALLD

Para ver la lista de tipos de conjuntos de IP soportados por **firewalld**, introduzca el siguiente comando como root.

```
~]# firewall-cmd --get-ipset-types
hash:ip hash:ip,mark hash:ip,port hash:ip,port,ip hash:ip,port,net hash:mac hash:net hash:net,iface
hash:net,net hash:net,port hash:net,port,net
```

5.15.1. Configuración de las opciones del conjunto IP mediante la CLI

Los conjuntos de IP se pueden utilizar en las zonas de **firewalld** como fuentes y también como fuentes en reglas ricas. En Red Hat Enterprise Linux, el método preferido es utilizar los conjuntos de IP creados con **firewalld** en una regla directa.

- Para listar los conjuntos de IP conocidos por **firewalld** en el entorno permanente, utilice el siguiente comando como **root**:

  ```
  # firewall-cmd --permanent --get-ipsets
  ```

- Para añadir un nuevo conjunto de IP, utilice el siguiente comando utilizando el entorno permanente como **root**:

  ```
  # firewall-cmd --permanent --new-ipset=test --type=hash:net
  success
  ```

  El comando anterior crea un nuevo conjunto IP con el nombre *test* y el tipo **hash:net** para **IPv4**. Para crear un conjunto de IP para usar con **IPv6**, añada la opción **--option=family=inet6**. Para que la nueva configuración sea efectiva en el entorno de ejecución, vuelva a cargar **firewalld**.

- Enumere el nuevo conjunto de IP con el siguiente comando como **root**:

  ```
  # firewall-cmd --permanent --get-ipsets
  test
  ```

- Para obtener más información sobre el conjunto de IP, utilice el siguiente comando como **root**:

  ```
  # firewall-cmd --permanent --info-ipset=test
  test
  type: hash:net
  options:
  entries:
  ```

  Tenga en cuenta que el conjunto de IP no tiene ninguna entrada en este momento.

- Para añadir una entrada al conjunto de IP de *test*, utilice el siguiente comando como **root**:

```
# firewall-cmd --permanent --ipset=test --add-entry=192.168.0.1
success
```

El comando anterior añade la dirección IP *192.168.0.1* al conjunto de IP.

- Para obtener la lista de entradas actuales en el conjunto de IP, utilice el siguiente comando como **root**:

```
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

- Generar un archivo con una lista de direcciones IP, por ejemplo:

```
# cat > iplist.txt <<EOL
192.168.0.2
192.168.0.3
192.168.1.0/24
192.168.2.254
EOL
```

El archivo con la lista de direcciones IP para un conjunto de IP debe contener una entrada por línea. Las líneas que comienzan con una almohadilla, un punto y coma o líneas vacías se ignoran.

- Para añadir las direcciones del archivo *iplist.txt*, utilice el siguiente comando como **root**:

```
# firewall-cmd --permanent --ipset=test --add-entries-from-file=iplist.txt
success
```

- Para ver la lista de entradas extendidas del conjunto de IP, utilice el siguiente comando como **root**:

```
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
192.168.0.2
192.168.0.3
192.168.1.0/24
192.168.2.254
```

- Para eliminar las direcciones del conjunto de IP y comprobar la lista de entradas actualizada, utilice los siguientes comandos como **root**:

```
# firewall-cmd --permanent --ipset=test --remove-entries-from-file=iplist.txt
success
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

- Puede añadir el conjunto de IPs como origen a una zona para gestionar todo el tráfico procedente de cualquiera de las direcciones listadas en el conjunto de IPs con una zona. Por ejemplo, para añadir el conjunto de IPs *test* como origen a la zona *drop* para descartar todos los paquetes procedentes de todas las entradas listadas en el conjunto de IPs *test*, utilice el siguiente comando como **root**:

```
# firewall-cmd --permanent --zone=drop --add-source=ipset:test
success
```

-

El prefijo **ipset:** en el origen muestra a **firewalld** que el origen es un conjunto de IP y no una dirección IP o un rango de direcciones.

Sólo la creación y eliminación de conjuntos de IP está limitada al entorno permanente, todas las demás opciones de conjuntos de IP pueden utilizarse también en el entorno de ejecución sin la opción **--permanent**.

AVISO

Red Hat no recomienda el uso de conjuntos de IP que no sean gestionados a través de **firewalld**. Para usar tales conjuntos de IP, se requiere una regla directa permanente para referenciar el conjunto, y se debe agregar un servicio personalizado para crear estos conjuntos de IP. Este servicio debe iniciarse antes de que se inicie firewalld, de lo contrario **firewalld** no podrá añadir las reglas directas que utilizan estos conjuntos. Puede añadir reglas directas permanentes con el archivo **/etc/firewalld/direct.xml**.

5.16. PRIORIZAR LAS REGLAS RICAS

Por defecto, las reglas ricas se organizan en base a su acción de regla. Por ejemplo, las reglas de **deny** tienen prioridad sobre las de **allow**. El parámetro **priority** en las reglas ricas proporciona a los administradores un control detallado sobre las reglas ricas y su orden de ejecución.

5.16.1. Cómo el parámetro de prioridad organiza las reglas en diferentes cadenas

Puede establecer el parámetro **priority** en una regla rica en cualquier número entre **-32768** y **32767**, y los valores más bajos tienen mayor precedencia.

El servicio **firewalld** organiza las reglas en función de su valor de prioridad en diferentes cadenas:

- Prioridad inferior a 0: la regla se redirige a una cadena con el sufijo **_pre**.
- Prioridad superior a 0: la regla se redirige a una cadena con el sufijo **_post**.
- Prioridad igual a 0: en función de la acción, la regla se redirige a una cadena con el **_log**, **_deny**, o **_allow** la acción.

Dentro de estas subcadenas, **firewalld** ordena las reglas en función de su valor de prioridad.

5.16.2. Establecer la prioridad de una regla rica

El procedimiento describe un ejemplo de cómo crear una regla rica que utiliza el parámetro **priority** para registrar todo el tráfico no permitido o denegado por otras reglas. Puede utilizar esta regla para marcar el tráfico inesperado.

Procedimiento

1. Añade una regla rica con una precedencia muy baja para registrar todo el tráfico que no haya sido igualado por otras reglas:

-

```
# firewall-cmd --add-rich-rule='rule priority=32767 log prefix="UNEXPECTED: " limit
value="5/m"'
```

Además, el comando limita el número de entradas de registro a **5** por minuto.

2. Opcionalmente, visualice la regla **nftables** que el comando del paso anterior creó:

```
# nft list chain inet firewalld filter_IN_public_post
table inet firewalld {
  chain filter_IN_public_post {
    log prefix "UNEXPECTED: " limit rate 5/minute
  }
}
```

5.17. CONFIGURACIÓN DEL BLOQUEO DEL CORTAFUEGOS

Las aplicaciones o servicios locales pueden cambiar la configuración del cortafuegos si se ejecutan como **root** (por ejemplo, **libvirt**). Con esta función, el administrador puede bloquear la configuración del cortafuegos para que ninguna aplicación o sólo las aplicaciones añadidas a la lista de permisos de bloqueo puedan solicitar cambios en el cortafuegos. La configuración de bloqueo está desactivada por defecto. Si está activada, el usuario puede estar seguro de que no se realizan cambios de configuración no deseados en el cortafuegos por parte de aplicaciones o servicios locales.

5.17.1. Configuración del bloqueo mediante la CLI

- Para consultar si el bloqueo está activado, utilice el siguiente comando como **root**:

```
# firewall-cmd --query-lockdown
```

El comando imprime **yes** con el estado de salida **0** si el bloqueo está activado. Imprime **no** con el estado de salida **1** en caso contrario.

- Para activar el bloqueo, introduzca el siguiente comando como **root**:

```
# firewall-cmd --lockdown-on
```

- Para desactivar el bloqueo, utilice el siguiente comando como **root**:

```
# firewall-cmd --lockdown-off
```

5.17.2. Configuración de las opciones de la lista de permisos de bloqueo mediante la CLI

La lista de permisos de bloqueo puede contener comandos, contextos de seguridad, usuarios e identificaciones de usuario. Si la entrada de un comando en la lista permitida termina con un asterisco "*", entonces todas las líneas de comando que comiencen con ese comando coincidirán. Si el asterisco "*" no está ahí, entonces el comando absoluto, incluyendo los argumentos, debe coincidir.

- El contexto es el contexto de seguridad (SELinux) de una aplicación o servicio en ejecución. Para obtener el contexto de una aplicación en ejecución utilice el siguiente comando:

```
$ ps -e --context
```

Este comando devuelve todas las aplicaciones en ejecución. Pase la salida por la herramienta **grep** para obtener la aplicación que le interesa. Por ejemplo:

```
$ ps -e --context | grep ejemplo_programa
```

- Para listar todas las líneas de comandos que están en la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-commands
```

- Para añadir un comando *command* a la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
```

- Para eliminar un comando *command* de la lista permitida, introduzca el siguiente comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-command='/usr/bin/python3 -Es
/usr/bin/command'
```

- Para consultar si el comando *command* está en la lista permitida, introduzca el siguiente comando como **root**:

```
# firewall-cmd --query-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
```

El comando imprime **yes** con el estado de salida **0** si es verdadero. Imprime **no** con el estado de salida **1** en caso contrario.

- Para listar todos los contextos de seguridad que están en la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-contexts
```

- Para añadir un contexto *context* a la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-context=contexto
```

- Para eliminar un contexto *context* de la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-context=contexto
```

- Para consultar si el contexto *context* está en la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --query-lockdown-whitelist-context=contexto
```

Imprime **yes** con el estado de salida **0**, si es verdadero, imprime **no** con el estado de salida **1** en caso contrario.

- Para listar todos los ID de usuario que están en la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-uids
```

- Para añadir un ID de usuario *uid* a la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-uid=uid
```

- Para eliminar un ID de usuario *uid* de la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-uid=uid
```

- Para consultar si el ID de usuario *uid* está en la lista de permitidos, introduzca el siguiente comando:

```
$ firewall-cmd --query-lockdown-whitelist-uid=uid
```

  Imprime **yes** con el estado de salida **0**, si es verdadero, imprime **no** con el estado de salida **1** en caso contrario.

- Para listar todos los nombres de usuario que están en la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-users
```

- Para añadir un nombre de usuario *user* a la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-user=user
```

- Para eliminar un nombre de usuario *user* de la lista de permitidos, introduzca el siguiente comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-user=user
```

- Para consultar si el nombre de usuario *user* está en la lista de permitidos, introduzca el siguiente comando:

```
$ firewall-cmd --query-lockdown-whitelist-user=user
```

  Imprime **yes** con el estado de salida **0**, si es verdadero, imprime **no** con el estado de salida **1** en caso contrario.

5.17.3. Configuración de las opciones de la lista de permisos de bloqueo mediante archivos de configuración

El archivo de configuración allowlist por defecto contiene el contexto **NetworkManager** y el contexto por defecto de **libvirt**. El ID de usuario 0 también está en la lista.

```
<?xml version="1.0" encoding="utf-8"?>
 <whitelist>
   <selinux context="system_u:system_r:NetworkManager_t:s0"/>
   <selinux context="system_u:system_r:virtd_t:s0-s0:c0.c1023"/>
   <user id="0"/>
 </whitelist>
```

A continuación se muestra un ejemplo de archivo de configuración allowlist que habilita todos los comandos de la utilidad **firewall-cmd**, para un usuario llamado *user* cuyo ID de usuario es **815**:

```
<?xml version="1.0" encoding="utf-8"?>
 <whitelist>
   <command name="/usr/libexec/platform-python -s /bin/firewall-cmd*"/>
   <selinux context="system_u:system_r:NetworkManager_t:s0"/>
   <user id="815"/>
   <user name="user"/>
 </whitelist>
```

Este ejemplo muestra tanto **user id** como **user name**, pero sólo se requiere una opción. Python es el intérprete y se antepone a la línea de comandos. También puede utilizar un comando específico, por ejemplo:

```
/usr/bin/python3 /bin/firewall-cmd --lockdown-on
```

En ese ejemplo, sólo se permite el comando **--lockdown-on**.

En Red Hat Enterprise Linux, todas las utilidades se colocan en el directorio **/usr/bin/** y el directorio **/bin/** está vinculado al directorio **/usr/bin/**. En otras palabras, aunque la ruta para **firewall-cmd** cuando se introduce como **root** podría resolver a **/bin/firewall-cmd**, ahora se puede utilizar **/usr/bin/firewall-cmd**. Todos los nuevos scripts deben utilizar la nueva ubicación. Pero tenga en cuenta que si los scripts que se ejecutan como **root** se escriben para utilizar la ruta **/bin/firewall-cmd**, entonces esa ruta de comandos debe añadirse en la lista de permisos además de la ruta **/usr/bin/firewall-cmd** que tradicionalmente sólo se utiliza para los usuarios que no son de**root**.

El ***** al final del atributo name de un comando significa que todos los comandos que comienzan con esta cadena coinciden. Si el ***** no está ahí, entonces el comando absoluto, incluyendo los argumentos, debe coincidir.

5.18. REGISTRO DE PAQUETES RECHAZADOS

Con la opción **LogDenied** en **firewalld**, es posible añadir un mecanismo de registro sencillo para los paquetes rechazados. Estos son los paquetes que son rechazados o descartados. Para cambiar la configuración del registro, edite el archivo **/etc/firewalld/firewalld.conf** o utilice la herramienta de configuración de la línea de comandos o de la interfaz gráfica de usuario.

Si **LogDenied** está habilitado, las reglas de registro se añaden justo antes de las reglas de rechazo y eliminación en las cadenas INPUT, FORWARD y OUTPUT para las reglas por defecto y también las reglas finales de rechazo y eliminación en las zonas. Los valores posibles para esta configuración son: **all** , **unicast**, **broadcast**, **multicast**, y **off**. El valor por defecto es **off**. Con la configuración **unicast**, **broadcast**, y **multicast**, la coincidencia **pkttype** se utiliza para coincidir con el tipo de paquete de la capa de enlace. Con **all**, se registran todos los paquetes.

Para listar la configuración real de **LogDenied** con firewall-cmd, utilice el siguiente comando como **root**:

```
# firewall-cmd --get-log-denied
off
```

Para cambiar la configuración de **LogDenied**, utilice el siguiente comando como **root**:

```
# firewall-cmd --set-log-denied=all
success
```

Para cambiar la configuración de **LogDenied** con la herramienta de configuración de la GUI **firewalld**, inicie **firewall-config**, haga clic en el menú **Options** y seleccione **Change Log Denied**. Aparece la ventana **LogDenied**. Seleccione la nueva configuración de **LogDenied** en el menú y haga clic en Aceptar.

5.19. INFORMACIÓN RELACIONADA

Las siguientes fuentes de información ofrecen recursos adicionales sobre **firewalld**.

Documentación instalada

- **firewalld(1)** página de manual - describe las opciones de comando para **firewalld**.
- **firewalld.conf(5)** página man - contiene información para configurar **firewalld**.
- **firewall-cmd(1)** página de manual - describe las opciones de comando para el cliente de línea de comandos **firewalld**.
- **firewall-config(1)** página de manual - describe la configuración de la **firewall-config** herramienta.
- **firewall-offline-cmd(1)** página de manual - describe las opciones de comando para el cliente de línea de comandos **firewalld** fuera de línea.
- **firewalld.icmptype(5)** página de manual - describe los archivos de configuración XML para el filtrado de **ICMP**.
- **firewalld.ipset(5)** man page - describe los archivos de configuración XML para los conjuntos **firewalld IP** .
- **firewalld.service(5)** - describe los archivos de configuración XML para **firewalld service**.
- **firewalld.zone(5)** man page - describe los archivos de configuración XML para la configuración de la zona **firewalld**.
- **firewalld.direct(5)** man page - describe el archivo de configuración de la interfaz directa **firewalld**.
- **firewalld.lockdown-whitelist(5)** man page - describe el archivo de configuración **firewalld** lockdown allowlist.
- **firewalld.richlanguage(5)** man page - describe la sintaxis de las reglas del lenguaje enriquecido **firewalld**.
- **firewalld.zones(5)** página man - descripción general de lo que son las zonas y cómo configurarlas.
- **firewalld.dbus(5)** man page - describe la interfaz **D-Bus** de **firewalld**.

Documentación en línea

- http://www.firewalld.org/ - **firewalld** página de inicio.

CAPÍTULO 6. INTRODUCCIÓN A NFTABLES

El marco de trabajo **nftables** proporciona facilidades de clasificación de paquetes y es el sucesor designado de las herramientas **iptables**, **ip6tables**, **arptables**, y **ebtables**. Ofrece numerosas mejoras en cuanto a comodidad, características y rendimiento con respecto a las anteriores herramientas de filtrado de paquetes, sobre todo:

- tablas de búsqueda en lugar de procesamiento lineal
- un único marco para los protocolos **IPv4** y **IPv6**
- reglas aplicadas atómicamente en lugar de buscar, actualizar y almacenar un conjunto completo de reglas
- soporte para la depuración y el rastreo en el conjunto de reglas (**nftrace**) y la supervisión de los eventos de rastreo (en la herramienta **nft**)
- sintaxis más coherente y compacta, sin extensiones específicas de protocolo
- una API Netlink para aplicaciones de terceros

Al igual que **iptables**, **nftables** utiliza tablas para almacenar cadenas. Las cadenas contienen reglas individuales para realizar acciones. La herramienta **nft** sustituye a todas las herramientas de los anteriores marcos de filtrado de paquetes. La biblioteca **libnftnl** puede utilizarse para la interacción de bajo nivel con **nftables** Netlink API sobre la biblioteca **libmnl**.

El efecto de los módulos en el conjunto de reglas **nftables** puede observarse utilizando el comando **nft** list rule set. Dado que estas herramientas añaden tablas, cadenas, reglas, conjuntos y otros objetos al conjunto de reglas de **nftables**, tenga en cuenta que las operaciones del conjunto de reglas de **nftables**, como el comando **nft flush ruleset**, podrían afectar a los conjuntos de reglas instalados mediante los comandos heredados anteriormente separados.

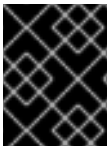
6.1. MIGRACIÓN DE IPTABLES A NFTABLES

Si ha actualizado su servidor a RHEL 8 o su configuración de cortafuegos todavía utiliza reglas de **iptables**, puede migrar sus reglas de **iptables** a **nftables**.

6.1.1. Cuándo utilizar firewalld, nftables o iptables

A continuación se presenta un breve resumen en el que se debe utilizar una de las siguientes utilidades:

- **firewalld**: Utilice la utilidad **firewalld** para casos de uso de cortafuegos sencillos. La utilidad es fácil de usar y cubre los casos de uso típicos para estos escenarios.
- **nftables**: Utilice la utilidad **nftables** para configurar cortafuegos complejos y de rendimiento crítico, como por ejemplo para toda una red.
- **iptables**: La utilidad **iptables** en Red Hat Enterprise Linux 8 utiliza la API del kernel **nf_tables** en lugar del back end **legacy**. La API **nf_tables** proporciona compatibilidad con versiones anteriores para que los scripts que utilizan comandos **iptables** sigan funcionando en Red Hat Enterprise Linux 8. Para los nuevos scripts de cortafuegos, Red Hat recomienda utilizar **nftables**.



IMPORTANTE

Para evitar que los diferentes servicios de firewall se influyan mutuamente, ejecute sólo uno de ellos en un host RHEL y desactive los demás servicios.

6.1.2. Conversión de reglas iptables a reglas nftables

Red Hat Enterprise Linux 8 proporciona las herramientas **iptables-translate** y **ip6tables-translate** para convertir las reglas existentes **iptables** o **ip6tables** en las equivalentes para **nftables**.

Tenga en cuenta que algunas extensiones no tienen soporte de traducción. Si existe una extensión de este tipo, la herramienta imprime la regla no traducida con el prefijo **#**. Por ejemplo:

```
# iptables-translate -A INPUT -j CHECKSUM --checksum-fill
nft # -A INPUT -j CHECKSUM --checksum-fill
```

Además, los usuarios pueden utilizar las herramientas **iptables-restore-translate** y **ip6tables-restore-translate** para traducir un volcado de reglas. Tenga en cuenta que antes de eso, los usuarios pueden utilizar los comandos **iptables-save** o **ip6tables-save** para imprimir un volcado de las reglas actuales. Por ejemplo:

```
# iptables-save >/tmp/iptables.dump
# iptables-restore-translate -f /tmp/iptables.dump

# Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
add table ip nat
...
```

Para obtener más información y una lista de posibles opciones y valores, introduzca el comando **iptables-translate --help**.

6.2. ESCRITURA Y EJECUCIÓN DE SCRIPTS NFTABLES

El marco de trabajo **nftables** proporciona un entorno de scripts nativo que aporta una gran ventaja sobre el uso de scripts de shell para mantener las reglas del cortafuegos: la ejecución de los scripts es atómica. Esto significa que el sistema aplica todo el script o impide la ejecución si se produce un error. Esto garantiza que el cortafuegos está siempre en un estado consistente.

Además, el entorno de scripts **nftables** permite a los administradores:

- añadir comentarios
- definir variables
- incluir otros archivos de conjuntos de reglas

En esta sección se explica cómo utilizar estas funciones, así como la creación y ejecución de los scripts de **nftables**.

Cuando se instala el paquete **nftables**, Red Hat Enterprise Linux crea automáticamente los scripts ***.nft** en el directorio **/etc/nftables/**. Estos scripts contienen comandos que crean tablas y cadenas vacías para diferentes propósitos. Usted puede extender estos archivos o escribir sus scripts.

6.2.1. La cabecera de script requerida en el script de nftables

Al igual que otros scripts, los scripts de **nftables** requieren una secuencia shebang en la primera línea del script que establece la directiva del intérprete.

Una secuencia de comandos **nftables** debe comenzar siempre con la siguiente línea:

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

IMPORTANTE

Si omite el parámetro **-f**, la utilidad **nft** no lee el script y muestra **Error: syntax error, unexpected newline, expecting string**.

6.2.2. Formatos de script de nftables soportados

El entorno de scripting **nftables** admite scripts en los siguientes formatos:

- Puede escribir una secuencia de comandos en el mismo formato que el comando **nft list ruleset** muestra el conjunto de reglas:

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

# Flush the rule set
flush ruleset

table inet example_table {
  chain example_chain {
    # Chain for incoming packets that drops all packets that
    # are not explicitly allowed by any rule in this chain
    type filter hook input priority 0; policy drop;

    # Accept connections to port 22 (ssh)
    tcp dport ssh accept
  }
}
```

- Puede utilizar la misma sintaxis para los comandos que en **nft**:

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

# Flush the rule set
flush ruleset

# Create a table
add table inet example_table

# Create a chain for incoming packets that drops all packets
# that are not explicitly allowed by any rule in this chain
add chain inet example_table example_chain { type filter hook input priority 0 ; policy drop ; }

# Add a rule that accepts connections to port 22 (ssh)
add rule inet example_table example_chain tcp dport ssh accept
```

6.2.3. Ejecución de scripts nftables

Para ejecutar un script de **nftables**, el script debe ser ejecutable. Sólo si el script está incluido en otro script, no es necesario que sea ejecutable. El procedimiento describe cómo hacer que un script sea ejecutable y ejecutar el script.

Requisitos previos

- El procedimiento de esta sección asume que usted almacenó un script **nftables** en el archivo **/etc/nftables/example_firewall.nft**.

Procedimiento

1. Pasos que se requieren sólo una vez:
   a. Opcionalmente, establezca el propietario del script en **root**:

   ```
   # chown root /etc/nftables/example_firewall.nft
   ```

   b. Haz que el script sea ejecutable para el propietario:

   ```
   # chmod u x /etc/nftables/example_firewall.nft
   ```

2. Ejecuta el script:

   ```
   # /etc/nftables/example_firewall.nft
   ```

   Si no se muestra ninguna salida, el sistema ha ejecutado el script con éxito.

IMPORTANTE

Incluso si **nft** ejecuta la secuencia de comandos con éxito, las reglas mal colocadas, los parámetros que faltan u otros problemas en la secuencia de comandos pueden hacer que el cortafuegos no se comporte como se espera.

Recursos adicionales

- Para más detalles sobre cómo establecer el propietario de un archivo, consulte la página de manual **chown(1)**.
- Para más detalles sobre cómo establecer los permisos de un archivo, consulte la página de manual **chmod(1)**.
- Sección 6.2.7, “Carga automática de las reglas de nftables al arrancar el sistema”

6.2.4. Uso de comentarios en los scripts de nftables

El entorno de scripting **nftables** interpreta todo lo que está a la derecha de un carácter **#** como un comentario.

Ejemplo 6.1. Comentarios en un script de nftables

Los comentarios pueden comenzar al principio de una línea, así como junto a un comando:

```
...
# Flush the rule set
```

```
flush ruleset

add table inet example_table  # Create a table
...
```

6.2.5. Uso de variables en un script de nftables

Para definir una variable en un script **nftables**, utilice la palabra clave **define**. Puede almacenar valores individuales y conjuntos anónimos en una variable. Para escenarios más complejos, utilice conjuntos o mapas de veredicto.

Variables con un solo valor
El siguiente ejemplo define una variable llamada **INET_DEV** con el valor **enp1s0**:

```
define INET_DEV = enp1s0
```

Puede utilizar la variable en el script escribiendo el signo **$** seguido del nombre de la variable:

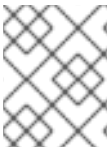
```
...
add rule inet example_table example_chain iifname $INET_DEV tcp dport ssh accept
...
```

Variables que contienen un conjunto anónimo
El siguiente ejemplo define una variable que contiene un conjunto anónimo:

```
define DNS_SERVERS = { 192.0.2.1, 192.0.2.2 }
```

Puede utilizar la variable en el script escribiendo el signo **$** seguido del nombre de la variable:

```
add rule inet ejemplo_tabla ejemplo_cadena ip daddr $DNS_SERVERS accept
```



NOTA

Tenga en cuenta que las llaves tienen una semántica especial cuando las utiliza en una regla porque indican que la variable representa un conjunto.

Recursos adicionales

- Para más detalles sobre los conjuntos, véase Sección 6.5, “Uso de conjuntos en los comandos de nftables”.
- Para más detalles sobre los mapas de veredicto, consulte Sección 6.6, “Uso de mapas de veredicto en los comandos de nftables”.

6.2.6. Inclusión de archivos en un script de nftables

El entorno de scripting **nftables** permite a los administradores incluir otros scripts utilizando la sentencia **include**.

Si especifica sólo un nombre de archivo sin una ruta absoluta o relativa, **nftables** incluye los archivos de la ruta de búsqueda por defecto, que está configurada en /**etc** en Red Hat Enterprise Linux.

Ejemplo 6.2. Incluir archivos del directorio de búsqueda por defecto

Para incluir un archivo del directorio de búsqueda por defecto:

```
include "ejemplo.nft"
```

Ejemplo 6.3. Incluir todos los archivos *.nft de un directorio

Para incluir todos los archivos que terminan en ***.nft** y que están almacenados en el directorio **/etc/nftables/rulesets/**:

```
include "/etc/nftables/rulesets/*.nft"
```

Tenga en cuenta que la sentencia **include** no coincide con los archivos que comienzan con un punto.

Recursos adicionales

- Para más detalles, consulte la sección **Include files** en la página de manual **nft(8)**.

6.2.7. Carga automática de las reglas de nftables al arrancar el sistema

El servicio **nftables** systemd carga los scripts del cortafuegos que se incluyen en el archivo **/etc/sysconfig/nftables.conf**. Esta sección explica cómo cargar las reglas del cortafuegos cuando el sistema se inicia.

Requisitos previos

- Los scripts de **nftables** se almacenan en el directorio **/etc/nftables/**.

Procedimiento

1. Edite el archivo **/etc/sysconfig/nftables.conf**.
   - Si mejora los scripts ***.nft** creados en **/etc/nftables/** cuando instaló el paquete **nftables**, descomente la sentencia **include** para estos scripts.
   - Si escribe scripts desde cero, añada las declaraciones **include** para incluir estos scripts. Por ejemplo, para cargar el script **/etc/nftables/*example*.nft** cuando se inicie el servicio **nftables**, añada

     ```
     include \ "/etc/nftables/example.nft"
     ```

2. Habilite el servicio **nftables**.

   ```
   # systemctl enable nftables
   ```

3. Opcionalmente, inicie el servicio **nftables** para cargar las reglas del cortafuegos sin reiniciar el sistema:

   ```
   # systemctl start nftables
   ```

Recursos adicionales

- Sección 6.2.2, “Formatos de script de nftables soportados”

6.3. CREACIÓN Y GESTIÓN DE TABLAS, CADENAS Y REGLAS NFTABLES

En esta sección se explica cómo visualizar los conjuntos de reglas de **nftables** y cómo gestionarlos.

6.3.1. Valores estándar de prioridad de la cadena y nombres textuales

Cuando se crea una cadena, en **priority** se puede establecer un valor entero o un nombre estándar que especifique el orden en el que atraviesan las cadenas con el mismo valor **hook**.

Los nombres y valores se definen en función de las prioridades que utiliza **xtables** al registrar sus cadenas por defecto.

NOTA

El comando **nft list chains** muestra por defecto valores de prioridad textuales. Puede ver el valor numérico pasando la opción **-y** al comando.

Ejemplo 6.4. Utilizar un valor textual para establecer la prioridad

El siguiente comando crea una cadena llamada **example_chain** en **example_table** utilizando el valor de prioridad estándar **50**:

 # nft add chain inet *example_table example_chain* { type *filter* hook *input* priority *50* \; policy accept \; }

Dado que la prioridad es un valor estándar, puede utilizar alternativamente el valor textual:

 # nft add chain inet *example_table example_chain* { type *filter* hook *input* priority *security* \; policy accept \; }

Tabla 6.1. Nombres de prioridad estándar, familia y matriz de compatibilidad de ganchos

Nombre	Valor	Familias	Ganchos
raw	-300	**ip**, **ip6**, **inet**	todo
mangle	-150	**ip**, **ip6**, **inet**	todo
dstnat	-100	**ip**, **ip6**, **inet**	prepago
filter	0	**ip**, **ip6**, **inet**, **arp**, **netdev**	todo
security	50	**ip**, **ip6**, **inet**	todo
srcnat	100	**ip**, **ip6**, **inet**	post ruta

Todas las familias utilizan los mismos valores, pero la familia **bridge** utiliza los siguientes valores:

Tabla 6.2. Nombres de prioridad estándar y compatibilidad de ganchos para la familia de puentes

Nombre	Valor	Ganchos
dstnat	-300	prepago
filter	-200	todo
out	100	salida
srcnat	300	post ruta

Recursos adicionales

- Para más detalles sobre otras acciones que puede ejecutar en las cadenas, consulte la sección **Chains** en la página de manual **nft(8)**.

6.3.2. Visualización de conjuntos de reglas nftables

Los conjuntos de reglas de **nftables** contienen tablas, cadenas y reglas. En esta sección se explica cómo visualizar estos conjuntos de reglas.

Procedimiento

- Para mostrar todos los conjuntos de reglas, introduzca:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport http accept
    tcp dport ssh accept
  }
}
```

NOTA

Por defecto, **nftables** no crea tablas previamente. Como consecuencia, al mostrar el conjunto de reglas en un host sin ninguna tabla, el comando **nft list ruleset** no muestra ninguna salida.

6.3.3. Creación de una tabla nftables

Una tabla en **nftables** es un espacio de nombres que contiene una colección de cadenas, reglas, conjuntos y otros objetos. Esta sección explica cómo crear una tabla.

Cada tabla debe tener definida una familia de direcciones. La familia de direcciones de una tabla define qué tipos de direcciones procesa la tabla. Puede establecer una de las siguientes familias de direcciones al crear una tabla:

- **ip**: Coincide sólo con los paquetes IPv4. Este es el valor por defecto si no se especifica una familia de direcciones.
- **ip6**: Coincide sólo con los paquetes IPv6.
- **inet**: Coincide con los paquetes IPv4 e IPv6.
- **arp**: Coincide con los paquetes del protocolo de resolución de direcciones IPv4 (ARP).
- **bridge**: Coincide con los paquetes que atraviesan un dispositivo de puente.
- **netdev**: Coincide con los paquetes de entrada.

Procedimiento

1. Utilice el comando **nft add table** para crear una nueva tabla. Por ejemplo, para crear una tabla llamada **example_table** que procese paquetes IPv4 e IPv6:

```
# nft add table inet example_table
```

2. Opcionalmente, se pueden listar todas las tablas del conjunto de reglas:

```
# nft list tables
table inet example_table
```

Recursos adicionales

- Para más detalles sobre las familias de direcciones, consulte la sección **Address families** en la página man **nft(8)**.
- Para más detalles sobre otras acciones que puede ejecutar en las tablas, consulte la sección **Tables** en la página de manual **nft(8)**.

6.3.4. Creación de una cadena nftables

Las cadenas son contenedores de reglas. Existen los siguientes dos tipos de reglas:

- Cadena base: Puedes utilizar las cadenas base como punto de entrada para los paquetes de la pila de red.
- Cadena regular: Puede utilizar las cadenas regulares como objetivo de **jump** y para organizar mejor las reglas.

El procedimiento describe cómo añadir una cadena base a una tabla existente.

Requisitos previos

- La tabla a la que se quiere añadir la nueva cadena existe.

Procedimiento

1. Utilice el comando **nft add chain** para crear una nueva cadena. Por ejemplo, para crear una cadena llamada **example_chain** en **example_table**:

```
# nft add chain inet example_table example_chain { type filter hook input priority 0 \\N; policy accept \N; }
```

IMPORTANTE

Para evitar que el shell interprete los puntos y comas como el final del comando, debe escapar los puntos y comas con una barra invertida.

Esta cadena filtra los paquetes entrantes. El parámetro **priority** especifica el orden en que **nftables** procesa las cadenas con el mismo valor de gancho. Un valor de prioridad más bajo tiene precedencia sobre los más altos. El parámetro **policy** establece la acción por defecto para las reglas de esta cadena. Tenga en cuenta que si está conectado al servidor de forma remota y establece la política por defecto en **drop**, se desconectará inmediatamente si ninguna otra regla permite el acceso remoto.

2. Opcionalmente, mostrar todas las cadenas:

```
# nft list chains
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
  }
}
```

Recursos adicionales

- Para más detalles sobre las familias de direcciones, consulte la sección **Address families** en la página man **nft(8)**.
- Para más detalles sobre otras acciones que puede ejecutar en las cadenas, consulte la sección **Chains** en la página de manual **nft(8)**.

6.3.5. Añadir una regla a una cadena nftables

Esta sección explica cómo añadir una regla a una cadena existente de **nftables**. Por defecto, el comando **nftables add rule** añade una nueva regla al final de la cadena.

Si, en cambio, desea insertar una regla al principio de la cadena, consulte Sección 6.3.6, “Insertar una regla en una cadena nftables”.

Requisitos previos

- La cadena a la que se quiere añadir la regla existe.

Procedimiento

1. Para añadir una nueva regla, utilice el comando **nft add rule**. Por ejemplo, para añadir una regla al **example_chain** en el **example_table** que permita el tráfico TCP en el puerto 22:

```
# nft add rule inet example_table example_chain tcp dport 22 accept
```

En lugar del número de puerto, puede especificar alternativamente el nombre del servicio. En el ejemplo, podría utilizar **ssh** en lugar del número de puerto **22**. Tenga en cuenta que un nombre de servicio se resuelve a un número de puerto basado en su entrada en el archivo **/etc/services**.

2. Opcionalmente, mostrar todas las cadenas y sus reglas en **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    ...
    tcp dport ssh accept
  }
}
```

Recursos adicionales

- Para más detalles sobre las familias de direcciones, consulte la sección **Address families** en la página man **nft(8)**.
- Para más detalles sobre otras acciones que puede ejecutar en las reglas, consulte la sección **Rules** en la página de manual **nft(8)**.

6.3.6. Insertar una regla en una cadena nftables

Esta sección explica cómo insertar una regla al principio de una cadena existente de **nftables** utilizando el comando **nftables insert rule**. Si, en cambio, desea añadir una regla al final de una cadena, consulte Sección 6.3.5, “Añadir una regla a una cadena nftables” .

Requisitos previos

- La cadena a la que se quiere añadir la regla existe.

Procedimiento

1. Para insertar una nueva regla, utilice el comando **nft insert rule**. Por ejemplo, para insertar una regla a la **example_chain** en el **example_table** que permite el tráfico TCP en el puerto 22:

```
# nft insert rule inet example_table example_chain tcp dport 22 accept
```

También puede especificar el nombre del servicio en lugar del número de puerto. En el ejemplo, podría utilizar **ssh** en lugar del número de puerto **22**. Tenga en cuenta que un nombre de servicio se resuelve a un número de puerto basado en su entrada en el archivo **/etc/services**.

2. Opcionalmente, mostrar todas las cadenas y sus reglas en **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept
    ...
  }
}
```

Recursos adicionales

- Para más detalles sobre las familias de direcciones, consulte la sección **Address families** en la página man **nft(8)**.
- Para más detalles sobre otras acciones que puede ejecutar en las reglas, consulte la sección **Rules** en la página de manual **nft(8)**.

6.4. CONFIGURACIÓN DE NAT CON NFTABLES

Con **nftables**, puede configurar los siguientes tipos de traducción de direcciones de red (NAT):

- Enmascaramiento
- NAT de origen (SNAT)
- NAT de destino (DNAT)

6.4.1. Los diferentes tipos de NAT: enmascaramiento, NAT de origen y NAT de destino

Estos son los diferentes tipos de traducción de direcciones de red (NAT):

Enmascaramiento y NAT de origen (SNAT)

Utilice uno de estos tipos de NAT para cambiar la dirección IP de origen de los paquetes. Por ejemplo, los proveedores de Internet no enrutan rangos de IP reservados, como **10.0.0.0/8**. Si utiliza rangos de IP reservadas en su red y los usuarios deben poder llegar a los servidores de Internet, asigne la dirección IP de origen de los paquetes de estos rangos a una dirección IP pública.
Tanto el enmascaramiento como el SNAT son muy similares. Las diferencias son:

- El enmascaramiento utiliza automáticamente la dirección IP de la interfaz saliente. Por lo tanto, utilice el enmascaramiento si la interfaz saliente utiliza una dirección IP dinámica.
- SNAT establece la dirección IP de origen de los paquetes a una IP especificada y no busca dinámicamente la IP de la interfaz de salida. Por lo tanto, SNAT es más rápido que el enmascaramiento. Utilice SNAT si la interfaz de salida utiliza una dirección IP fija.

NAT de destino (DNAT)

Utiliza este tipo de NAT para redirigir el tráfico entrante a un host diferente. Por ejemplo, si su servidor web utiliza una dirección IP de un rango IP reservado y, por tanto, no es accesible directamente desde Internet, puede establecer una regla DNAT en el router para redirigir el tráfico entrante a este servidor.

6.4.2. Configuración del enmascaramiento mediante nftables

El enmascaramiento permite a un router cambiar dinámicamente la IP de origen de los paquetes enviados a través de una interfaz por la dirección IP de la misma. Esto significa que si la interfaz recibe una nueva IP asignada, **nftables** utiliza automáticamente la nueva IP al sustituir la IP de origen.

El siguiente procedimiento describe cómo sustituir la IP de origen de los paquetes que salen del host a través de la interfaz **ens3** por la IP establecida en **ens3**.

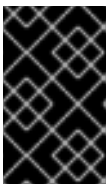
Procedimiento

1. Crea una tabla:

```
# nft add table nat
```

2. Añade las cadenas **prerouting** y **postrouting** a la tabla:

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```



IMPORTANTE

Incluso si no se añade una regla a la cadena **prerouting**, el marco **nftables** requiere esta cadena para que coincida con las respuestas de los paquetes entrantes.

Tenga en cuenta que debe pasar la opción **--** al comando **nft** para evitar que el shell interprete el valor de prioridad negativo como una opción del comando **nft**.

3. Añada una regla a la cadena **postrouting** que coincida con los paquetes salientes en la interfaz **ens3**:

```
# nft add rule nat postrouting oifname "ens3" masquerade
```

6.4.3. Configuración del NAT de origen mediante nftables

En un router, Source NAT (SNAT) permite cambiar la IP de los paquetes enviados a través de una interfaz a una dirección IP específica.

El siguiente procedimiento describe cómo sustituir la IP de origen de los paquetes que salen del router a través de la interfaz **ens3** a **192.0.2.1**.

Procedimiento

1. Crea una tabla:

```
# nft add table nat
```

2. Añade las cadenas **prerouting** y **postrouting** a la tabla:

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

IMPORTANTE

Incluso si no se añade una regla a la cadena **postrouting**, el marco **nftables** requiere esta cadena para que coincida con las respuestas de los paquetes salientes.

Tenga en cuenta que debe pasar la opción **--** al comando **nft** para evitar que el shell interprete el valor de prioridad negativo como una opción del comando **nft**.

3. Añada una regla a la cadena **postrouting** que sustituya la IP de origen de los paquetes salientes a través de **ens3** por **192.0.2.1**:

```
# nft add rule nat postrouting oifname "ens3" snat to 192.0.2.1
```

Recursos adicionales

- Sección 6.7.2, "Reenvío de paquetes entrantes en un puerto local específico a un host diferente"

6.4.4. Configuración del NAT de destino mediante nftables

La NAT de destino permite redirigir el tráfico en un router a un host al que no se puede acceder directamente desde Internet.

El siguiente procedimiento describe cómo redirigir el tráfico entrante enviado al puerto **80** y **443** del router al host con la dirección IP **192.0.2.1**.

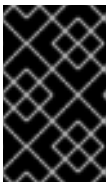
Procedimiento

1. Crea una tabla:

```
# nft add table nat
```

2. Añade las cadenas **prerouting** y **postrouting** a la tabla:

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```



IMPORTANTE

Incluso si no se añade una regla a la cadena **postrouting**, el marco **nftables** requiere esta cadena para que coincida con las respuestas de los paquetes salientes.

Tenga en cuenta que debe pasar la opción **--** al comando **nft** para evitar que el shell interprete el valor de prioridad negativo como una opción del comando **nft**.

3. Añada una regla a la cadena **prerouting** que redirija el tráfico entrante en la interfaz **ens3** enviado al puerto **80** y **443** al host con la IP **192.0.2.1**:

```
# nft add rule nat prerouting iifname ens3 tcp dport { 80, 443 } dnat to 192.0.2.1
```

4. Dependiendo de su entorno, añada una regla SNAT o de enmascaramiento para cambiar la dirección de origen:

   a. Si la interfaz **ens3** utiliza direcciones IP dinámicas, añada una regla de enmascaramiento:

```
# nft add rule nat postrouting oifname \ "ens3" masquerade
```

   b. Si la interfaz **ens3** utiliza una dirección IP estática, añada una regla SNAT. Por ejemplo, si el **ens3** utiliza la dirección IP **198.51.100.1**:

```
nft add rule nat postrouting oifname \ "ens3" snat to 198.51.100.1
```

Recursos adicionales

- Sección 6.4.1, "Los diferentes tipos de NAT: enmascaramiento, NAT de origen y NAT de destino"

6.5. USO DE CONJUNTOS EN LOS COMANDOS DE NFTABLES

El marco de trabajo **nftables** admite de forma nativa los conjuntos. Puedes utilizar conjuntos, por ejemplo, si una regla debe coincidir con varias direcciones IP, números de puerto, interfaces o cualquier otro criterio de coincidencia.

6.5.1. Uso de conjuntos anónimos en nftables

Un conjunto anónimo contiene valores separados por comas y encerrados entre corchetes, como **{ 22, 80, 443 }**, que se utilizan directamente en una regla. También puede utilizar conjuntos anónimos para direcciones IP o cualquier otro criterio de coincidencia.

El inconveniente de los conjuntos anónimos es que si se quiere cambiar el conjunto, hay que sustituir la regla. Para una solución dinámica, utilice conjuntos con nombre como se describe en Sección 6.5.2, "Uso de conjuntos con nombre en nftables".

Requisitos previos

- Existe la cadena **example_chain** y la tabla **example_table** en la familia **inet**.

Procedimiento

1. Por ejemplo, para añadir una regla a **example_chain** en **example_table** que permita el tráfico entrante al puerto **22**, **80** y **443**:

```
# nft add rule inet example_table example_chain tcp dport { 22, 80, 443 } accept
```

2. Opcionalmente, mostrar todas las cadenas y sus reglas en **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport { ssh, http, https } accept
  }
}
```

6.5.2. Uso de conjuntos con nombre en nftables

El marco **nftables** admite conjuntos con nombre mutables. Un conjunto con nombre es una lista o rango de elementos que se puede utilizar en múltiples reglas dentro de una tabla. Otra ventaja con respecto a los conjuntos anónimos es que se puede actualizar un conjunto con nombre sin sustituir las reglas que lo utilizan.

Cuando se crea un conjunto con nombre, se debe especificar el tipo de elementos que contiene el conjunto. Puede establecer los siguientes tipos:

- **ipv4_addr** para un conjunto que contiene direcciones o rangos IPv4, como **192.0.2.1** o **192.0.2.0/24**.
- **ipv6_addr** para un conjunto que contiene direcciones o rangos IPv6, como **2001:db8:1::1** o **2001:db8:1::1/64**.
- **ether_addr** para un conjunto que contiene una lista de direcciones de control de acceso al medio (MAC), como **52:54:00:6b:66:42**.
- **inet_proto** para un conjunto que contiene una lista de tipos de protocolo de Internet, como **tcp**.
- **inet_service** para un conjunto que contiene una lista de servicios de Internet, como **ssh**.
- **mark** para un conjunto que contiene una lista de marcas de paquetes. Las marcas de paquetes pueden ser cualquier valor entero positivo de 32 bits (**0** a **2147483647**].

Requisitos previos

- La cadena **example_chain** y la tabla **example_table** existen.

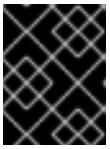
Procedimiento

1. Cree un conjunto vacío. Los siguientes ejemplos crean un conjunto para direcciones IPv4:
   - Para crear un conjunto que pueda almacenar varias direcciones IPv4 individuales:

     ```
     # nft add set inet example_table example_set { type ipv4_addr \; }
     ```

   - Para crear un conjunto que pueda almacenar rangos de direcciones IPv4:

     ```
     # nft add set inet example_table example_set { type ipv4_addr \; flags interval \; }
     ```

   IMPORTANTE

   Para evitar que el shell interprete los puntos y comas como el final del comando, debe escapar los puntos y comas con una barra invertida.

2. Opcionalmente, cree reglas que utilicen el conjunto. Por ejemplo, el siguiente comando agrega una regla a la **example_chain** en el **example_table** que descartará todos los paquetes de las direcciones IPv4 en **example_set**.

   ```
   # nft add rule inet example_table example_chain ip saddr @example_set drop
   ```

   Como **example_set** está todavía vacío, la regla no tiene actualmente ningún efecto.

3. Añadir direcciones IPv4 a **example_set**:
   - Si crea un conjunto que almacena direcciones IPv4 individuales, introduzca:

     ```
     # nft add element inet example_table example_set { 192.0.2.1, 192.0.2.2 }
     ```

   - Si crea un conjunto que almacena rangos IPv4, introduzca:

     ```
     # nft add element inet example_table example_set { 192.0.2.0-192.0.2.255 }
     ```

Cuando se especifica un rango de direcciones IP, se puede utilizar alternativamente la notación Classless Inter-Domain Routing (CIDR), como **192.0.2.0/24** en el ejemplo anterior.

6.5.3. Información relacionada

- Para más detalles sobre los conjuntos, consulte la sección **Sets** en la página de manual **nft(8)**.

6.6. USO DE MAPAS DE VEREDICTO EN LOS COMANDOS DE NFTABLES

Los mapas de veredicto, también conocidos como diccionarios, permiten a **nft** realizar una acción basada en la información del paquete mediante la asignación de criterios de coincidencia a una acción.

6.6.1. Uso de mapas literales en nftables

Un mapa literal es una **{ *match_criteria* : *action* }** que se utiliza directamente en una regla. La sentencia puede contener múltiples mapeos separados por comas.

El inconveniente de un mapa literal es que si se quiere cambiar el mapa, hay que sustituir la regla. Para una solución dinámica, utilice mapas de veredicto con nombre como se describe en Sección 6.6.2, “Uso de mapas de veredicto mutables en nftables”.

El ejemplo describe cómo utilizar un mapa literal para enrutar paquetes TCP y UDP del protocolo IPv4 e IPv6 a diferentes cadenas para contar los paquetes TCP y UDP entrantes por separado.

Procedimiento

1. Cree la página web **example_table**:

```
# nft add table inet example_table
```

2. Cree la cadena **tcp_packets** en **example_table**:

```
# nft add chain inet example_table tcp_packets
```

3. Agregue una regla a **tcp_packets** que cuente el tráfico en esta cadena:

```
# nft add rule inet example_table tcp_packets counter
```

4. Cree la cadena **udp_packets** en **example_table**

```
# nft add chain inet example_table udp_packets
```

5. Agregue una regla a **udp_packets** que cuente el tráfico en esta cadena:

```
# nft add rule inet example_table udp_packets counter
```

6. Cree una cadena para el tráfico entrante. Por ejemplo, para crear una cadena llamada **incoming_traffic** en **example_table** que filtre el tráfico entrante:

```
# nft add chain inet example_table incoming_traffic { type filter hook input priority 0 \; }
```

7. Añade una regla con un mapa literal a **incoming_traffic**:

```
# nft add rule inet example_table incoming_traffic ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
```

El mapa literal distingue los paquetes y los envía a las diferentes cadenas de contadores en función de su protocolo.

8. Para listar los contadores de tráfico, visualice **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain tcp_packets {
    counter packets 36379 bytes 2103816
  }

  chain udp_packets {
    counter packets 10 bytes 1559
  }

  chain incoming_traffic {
    type filter hook input priority filter; policy accept;
    ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
  }
}
```

Los contadores de la cadena **tcp_packets** y **udp_packets** muestran tanto el número de paquetes como de bytes recibidos.

6.6.2. Uso de mapas de veredicto mutables en nftables

El marco **nftables** admite mapas de veredicto mutables. Puedes utilizar estos mapas en múltiples reglas dentro de una tabla. Otra ventaja con respecto a los mapas literales es que puedes actualizar un mapa mutable sin reemplazar las reglas que lo utilizan.

Cuando se crea un mapa de veredicto mutable, se debe especificar el tipo de elementos

- **ipv4_addr** para un mapa cuya parte coincidente contiene una dirección IPv4, como **192.0.2.1**.
- **ipv6_addr** para un mapa cuya parte coincidente contiene una dirección IPv6, como **2001:db8:1::1**.
- **ether_addr** para un mapa cuya parte coincidente contiene una dirección de control de acceso al medio (MAC), como **52:54:00:6b:66:42**.
- **inet_proto** para un mapa cuya parte coincidente contiene un tipo de protocolo de Internet, como **tcp**.
- **inet_service** para un mapa cuya parte de coincidencia contiene un número de puerto de nombre de servicios de Internet, como **ssh** o **22**.
- **mark** para un mapa cuya parte coincidente contiene una marca de paquete. Una marca de paquete puede ser cualquier valor entero positivo de 32 bits (**0** a **2147483647**.
- **counter** para un mapa cuya parte de coincidencia contiene un valor de contador. El valor del contador puede ser cualquier valor entero positivo de 64 bits.

- **quota** para un mapa cuya parte de coincidencia contiene un valor de cuota. El valor de la cuota puede ser cualquier valor entero positivo de 64 bits.

El ejemplo describe cómo permitir o descartar paquetes entrantes basándose en su dirección IP de origen. Utilizando un mapa de veredicto mutable, sólo se requiere una única regla para configurar este escenario, mientras que las direcciones IP y las acciones se almacenan dinámicamente en el mapa. El procedimiento también describe cómo añadir y eliminar entradas del mapa.

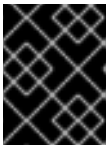
Procedimiento

1. Cree una tabla. Por ejemplo, para crear una tabla llamada **example_table** que procese paquetes IPv4:

```
# nft add table ip example_table
```

2. Cree una cadena. Por ejemplo, para crear una cadena llamada **example_chain** en **example_table**:

```
# nft add chain ip example_table example_chain { type filter hook input priority 0 \_; }
```



IMPORTANTE

Para evitar que el shell interprete los puntos y comas como el final del comando, debe escapar los puntos y comas con una barra invertida.

3. Cree un mapa vacío. Por ejemplo, para crear un mapa de direcciones IPv4:

```
# nft add map ip example_table example_map { type ipv4_addr : verdict \; }
```

4. Cree reglas que utilicen el mapa. Por ejemplo, el siguiente comando añade una regla a **example_chain** en **example_table** que aplica acciones a las direcciones IPv4 que están definidas en **example_map**:

```
# nft add rule example_table example_chain ip saddr vmap @example_map
```

5. Agregue las direcciones IPv4 y las acciones correspondientes a **example_map**:

```
# nft add element ip example_table example_map { 192.0.2.1 : accept, 192.0.2.2 : drop }
```

Este ejemplo define las asignaciones de direcciones IPv4 a acciones. En combinación con la regla creada anteriormente, el cortafuegos acepta los paquetes de **192.0.2.1** y los descarta de **192.0.2.2**.

6. Opcionalmente, puede mejorar el mapa añadiendo otra dirección IP y una declaración de acción:

```
# nft add element ip example_table example_map { 192.0.2.3 : accept }
```

7. Opcionalmente, eliminar una entrada del mapa:

```
# nft delete element ip example_table example_map { 192.0.2.1 }
```

8. Opcionalmente, mostrar el conjunto de reglas:

```
# nft list ruleset
table ip example_table {
  map example_map {
    type ipv4_addr : verdict
    elements = { 192.0.2.2 : drop, 192.0.2.3 : accept }
  }

  chain example_chain {
    type filter hook input priority filter; policy accept;
    ip saddr vmap @example_map
  }
}
```

6.6.3. Información relacionada

- Para más detalles sobre los mapas de veredicto, consulte la sección **Maps** en la página de manual **nft(8)**.

6.7. CONFIGURACIÓN DEL REENVÍO DE PUERTOS MEDIANTE NFTABLES

El reenvío de puertos permite a los administradores reenviar los paquetes enviados a un puerto de destino específico a un puerto local o remoto diferente.

Por ejemplo, si su servidor web no tiene una dirección IP pública, puede establecer una regla de reenvío de puertos en su cortafuegos que reenvíe los paquetes entrantes en los puertos **80** y **443** del cortafuegos al servidor web. Con esta regla del cortafuegos, los usuarios de Internet pueden acceder al servidor web utilizando la IP o el nombre de host del cortafuegos.

6.7.1. Reenvío de paquetes entrantes a un puerto local diferente

Esta sección describe un ejemplo de cómo reenviar paquetes IPv4 entrantes en el puerto **8022** al puerto **22** en el sistema local.

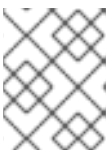
Procedimiento

1. Cree una tabla llamada **nat** con la familia de direcciones **ip**:

```
# nft add table ip nat
```

2. Añade las cadenas **prerouting** y **postrouting** a la tabla:

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
```



NOTA

Pase la opción **--** al comando **nft** para evitar que el shell interprete el valor de prioridad negativo como una opción del comando **nft**.

3. Añade una regla a la cadena **prerouting** que redirige los paquetes entrantes en el puerto **8022** al puerto local **22**:

```
# nft add rule ip nat prerouting tcp dport 8022 redirect to :22
```

6.7.2. Reenvío de paquetes entrantes en un puerto local específico a un host diferente

Puede utilizar una regla de traducción de direcciones de red de destino (DNAT) para reenviar los paquetes entrantes en un puerto local a un host remoto. Esto permite a los usuarios de Internet acceder a un servicio que se ejecuta en un host con una dirección IP privada.

El procedimiento describe cómo reenviar los paquetes IPv4 entrantes en el puerto local **443** al mismo número de puerto en el sistema remoto con la dirección IP **192.0.2.1**.

Requisito previo

- Usted está conectado como el usuario **root** en el sistema que debe reenviar los paquetes.

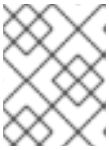
Procedimiento

1. Cree una tabla llamada **nat** con la familia de direcciones **ip**:

```
# nft add table ip nat
```

2. Añade las cadenas **prerouting** y **postrouting** a la tabla:

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
```



NOTA

Pase la opción **--** al comando **nft** para evitar que el shell interprete el valor de prioridad negativo como una opción del comando **nft**.

3. Añade una regla a la cadena **prerouting** que redirige los paquetes entrantes en el puerto **443** al mismo puerto en **192.0.2.1**:

```
# nft add rule ip nat prerouting tcp dport 443 dnat to 192.0.2.1
```

4. Añade una regla a la cadena **postrouting** para enmascarar el tráfico saliente:

```
# nft add rule ip daddr 192.0.2.1 masquerade
```

5. Activar el reenvío de paquetes:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

6.8. USO DE NFTABLES PARA LIMITAR LA CANTIDAD DE CONEXIONES

Puede utilizar **nftables** para limitar el número de conexiones o para bloquear las direcciones IP que intenten establecer una cantidad determinada de conexiones para evitar que utilicen demasiados recursos del sistema.

6.8.1. Limitación del número de conexiones mediante nftables

El parámetro **ct count** de la utilidad **nft** permite a los administradores limitar el número de conexiones. El procedimiento describe un ejemplo básico de cómo limitar las conexiones entrantes.

Requisitos previos

- La base **example_chain** en **example_table** existe.

Procedimiento

1. Añade una regla que permita sólo dos conexiones simultáneas al puerto SSH (22) desde una dirección IPv4 y rechace todas las demás conexiones desde la misma IP:

```
# nft add rule ip example_table example_chain tcp dport ssh meter example_meter { ip saddr ct count over 2 } counter reject
```

2. Opcionalmente, visualice el contador creado en el paso anterior:

```
# nft list meter ip example_table example_meter
table ip example_table {
  meter example_meter {
    type ipv4_addr
    size 65535
    elements = { 192.0.2.1 : ct count over 2 , 192.0.2.2 : ct count over 2  }
  }
}
```

La entrada **elements** muestra las direcciones que actualmente coinciden con la regla. En este ejemplo, **elements** muestra las direcciones IP que tienen conexiones activas al puerto SSH. Tenga en cuenta que la salida no muestra el número de conexiones activas o si las conexiones fueron rechazadas.

6.8.2. Bloqueo de direcciones IP que intentan más de diez nuevas conexiones TCP entrantes en un minuto

El marco **nftables** permite a los administradores actualizar dinámicamente los conjuntos. Esta sección explica cómo utilizar esta función para bloquear temporalmente los hosts que establecen más de diez conexiones TCP IPv4 en un minuto. Después de cinco minutos, **nftables** elimina automáticamente la dirección IP de la lista de denegación.

Procedimiento

1. Crear la tabla **filter** con la familia de direcciones **ip**:

```
# nft add table ip filter
```

2. Añade la cadena **input** a la tabla **filter**:

```
# nft add chain ip filter input { type filter hook input priority 0 \; }
```

3. Añade un conjunto llamado **denylist** a la tabla **filter**:

```
# nft add set ip filter denylist { type ipv4_addr \ ; flags dynamic, timeout \ ; timeout 5m \ ; }
```

Este comando crea un conjunto dinámico de direcciones IPv4. El parámetro **timeout 5m** define que **nftables** elimine automáticamente las entradas después de 5 minutos del conjunto.

4. Añada una regla que añada automáticamente la dirección IP de origen de los hosts que intenten establecer más de diez nuevas conexiones TCP en un minuto al conjunto **denylist**:

```
# nft add rule ip filter input ip protocol tcp ct state new, untracked limit rate over 10/minute add @denylist { ip saddr }
```

5. Añada una regla que elimine todas las conexiones de las direcciones IP del conjunto **denylist**:

```
# nft add rule ip filter input ip saddr @denylist drop
```

Recursos adicionales

- Sección 6.5.2, "Uso de conjuntos con nombre en nftables"

6.9. DEPURACIÓN DE LAS REGLAS DE NFTABLES

El marco **nftables** proporciona diferentes opciones para que los administradores puedan depurar las reglas y si los paquetes coinciden con ellas. Esta sección describe estas opciones.

6.9.1. Crear una regla con un contador

Para identificar si una regla coincide, puede utilizar un contador. Esta sección describe cómo crear una nueva regla con un contador.

Para un procedimiento que añade un contador a una regla existente, véase Sección 6.9.2, "Añadir un contador a una regla existente".

Requisitos previos

- La cadena a la que se quiere añadir la regla existe.

Procedimiento

1. Añada una nueva regla con el parámetro **counter** a la cadena. El siguiente ejemplo añade una regla con un contador que permite el tráfico TCP en el puerto 22 y cuenta los paquetes y el tráfico que coinciden con esta regla:

```
# nft add rule inet example_table example_chain tcp dport 22 counter accept
```

2. Para visualizar los valores del contador:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
```

```
    tcp dport ssh counter packets 6872 bytes 105448565 accept
  }
}
```

6.9.2. Añadir un contador a una regla existente

Para identificar si una regla coincide, puede utilizar un contador. Esta sección describe cómo añadir un contador a una regla existente.

Para conocer el procedimiento para añadir una nueva regla con un contador, consulte Sección 6.9.1, "Crear una regla con un contador".

Requisitos previos

- La regla a la que se quiere añadir el contador existe.

Procedimiento

1. Muestra las reglas de la cadena incluyendo sus asas:

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept # handle 4
  }
}
```

2. Añada el contador sustituyendo la regla pero con el parámetro **counter**. El siguiente ejemplo sustituye la regla mostrada en el paso anterior y añade un contador:

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 counter accept
```

3. Para visualizar los valores del contador:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport ssh counter packets 6872 bytes 105448565 accept
  }
}
```

6.9.3. Supervisión de los paquetes que coinciden con una regla existente

La función de rastreo en **nftables** en combinación con el comando **nft monitor** permite a los administradores mostrar los paquetes que coinciden con una regla. El procedimiento describe cómo habilitar el rastreo para una regla, así como la supervisión de los paquetes que coinciden con esta regla.

Requisitos previos

- La regla a la que se quiere añadir el contador existe.

Procedimiento

1. Muestra las reglas de la cadena incluyendo sus asas:

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept # handle 4
  }
}
```

2. Añada la función de rastreo sustituyendo la regla pero con los parámetros de **meta nftrace set 1**. El siguiente ejemplo reemplaza la regla mostrada en el paso anterior y habilita el rastreo:

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 meta nftrace set 1 accept
```

3. Utilice el comando **nft monitor** para mostrar el rastreo. El siguiente ejemplo filtra la salida del comando para mostrar sólo las entradas que contienen **inet example_table example_chain**:

```
# nft monitor | grep "inet example_table example_chain"
trace id 3c5eb15e inet example_table example_chain packet: iif "enp1s0" ether saddr 52:54:00:17:ff:e4 ether daddr 52:54:00:72:2f:6e ip saddr 192.0.2.1 ip daddr 192.0.2.2 ip dscp cs0 ip ecn not-ect ip ttl 64 ip id 49710 ip protocol tcp ip length 60 tcp sport 56728 tcp dport ssh tcp flags == syn tcp window 64240
trace id 3c5eb15e inet example_table example_chain rule tcp dport ssh nftrace set 1 accept (verdict accept)
...
```

AVISO

Dependiendo del número de reglas con el rastreo activado y de la cantidad de tráfico que coincida, el comando **nft monitor** puede mostrar una gran cantidad de salida. Utilice **grep** u otras utilidades para filtrar la salida.

6.10. COPIA DE SEGURIDAD Y RESTAURACIÓN DE CONJUNTOS DE REGLAS NFTABLES

Esta sección describe cómo hacer una copia de seguridad de las reglas de **nftables** en un archivo, así como la restauración de las reglas desde un archivo.

Los administradores pueden utilizar un archivo con las reglas para, por ejemplo, transferirlas a un servidor diferente.

6.10.1. Copia de seguridad de los conjuntos de reglas de nftables en un archivo

Esta sección describe cómo hacer una copia de seguridad de los conjuntos de reglas de **nftables** en un archivo.

Procedimiento

- Para hacer una copia de seguridad de las reglas de **nftables**:
  - En formato **nft list ruleset**:

```
# nft list ruleset > file.nft
```

  - En formato JSON:

```
# nft -j list ruleset > file.json
```

6.10.2. Restauración de conjuntos de reglas nftables desde un archivo

Esta sección describe cómo restaurar los conjuntos de reglas de **nftables**.

Procedimiento

- Para restaurar las reglas de **nftables**:
  - Si el archivo a restaurar está en formato **nft list ruleset** o contiene comandos **nft**:

```
# nft -f file.nft
```

  - Si el archivo a restaurar está en formato JSON:

```
# nft -j -f file.json
```

6.11. INFORMACIÓN RELACIONADA

- La entrada del blog Using nftables in Red Hat Enterprise Linux 8 proporciona una visión general sobre el uso de las características de **nftables**.
- ¿Qué viene después de iptables? Su sucesor, por supuesto: el artículo nftables explica por qué **nftables** sustituye a **iptables**.
- El artículo Firewalld: The Future is nftables proporciona información adicional sobre **nftables** como back end por defecto para **firewalld**.