ASICI

Asociación Internacional de Ciberseguridad

Guía Comandos del Firewall UFW (Uncomplicated Firewall)

UFW (Uncomplicated Firewall)

UFW (Uncomplicated Firewall) es la interfaz de línea de comandos predeterminada para gestionar `netfilter` (el framework de firewall del kernel de Linux) en distribuciones basadas en Debian/Ubuntu. Como su nombre indica, UFW está diseñado para simplificar la configuración del firewall, haciéndola accesible incluso para usuarios con menos experiencia en `iptables`. Es una herramienta esencial para asegurar cualquier servidor o estación de trabajo Linux.

1. ¿Qué es UFW y por qué usarlo?

UFW es una capa de abstracción sobre `iptables`. En lugar de tener que lidiar con las complejas cadenas y reglas de `iptables`, UFW ofrece una sintaxis mucho más intuitiva para definir qué tráfico se permite y cuál se bloquea.

Ventajas de UFW:

- **Simplicidad:** Configurar un firewall es mucho más fácil y rápido.
- **Fácil de entender:** Las reglas son legibles y lógicas.
- **Seguridad por defecto:** Por defecto, bloquea todo el tráfico entrante a menos que se especifique lo contrario.
- **Integración:** Se integra bien con el sistema de inicio de Linux y persiste las reglas a través de los reinicios.

2. Estados Básicos de UFW

Antes de configurar reglas, es importante conocer el estado actual de UFW.

- **`ufw status`:** Muestra el estado actual del firewall (activo/inactivo) y las reglas configuradas.
    - Ejemplo: `sudo ufw status`
    - `sudo ufw status verbose`: Muestra más detalles, incluyendo la política por defecto.
    - `sudo ufw status numbered`: Muestra las reglas con números, útil para eliminarlas por índice.
- **`ufw enable`:** Activa el firewall. **¡Advertencia!** Al habilitarlo, UFW por defecto bloquea todas las conexiones entrantes y permite todas las salientes. Asegúrate de haber permitido el acceso SSH (puerto 22) antes de habilitarlo si estás conectado remotamente, de lo contrario, podrías perder el acceso.
    - Ejemplo: `sudo ufw enable`
- **`ufw disable`:** Desactiva el firewall.
    - Ejemplo: `sudo ufw disable`
- **`ufw reset`:** Restablece el firewall a sus valores por defecto (elimina todas las reglas y deshabilita UFW). **¡Cuidado!**
    - Ejemplo: `sudo ufw reset`

3. Políticas por Defecto (Default Policies)

Las políticas por defecto definen el comportamiento de UFW para el tráfico que no coincide con ninguna regla explícita.

- **`ufw default deny incoming`:** Bloquea todo el tráfico entrante por defecto (es la política recomendada y por defecto de UFW).
    - Ejemplo: `sudo ufw default deny incoming`
- **`ufw default allow outgoing`:** Permite todo el tráfico saliente por defecto (es la política recomendada y por defecto de UFW).
    - Ejemplo: `sudo ufw default allow outgoing`
- **`ufw default allow incoming`:** Permite todo el tráfico entrante por defecto. **NO RECOMENDADO** para entornos de producción.
    - Ejemplo: `sudo ufw default allow incoming`

4. Añadir Reglas (Allowing/Denying Traffic)

Las reglas permiten o deniegan tráfico específico basado en puertos, protocolos, direcciones IP, etc.

Por Puerto y Protocolo:

- **`ufw allow <puerto>/<protocolo>`:** Permite el tráfico a un puerto específico y protocolo.
    - Ejemplo (permitir SSH): `sudo ufw allow 22/tcp`
    - Ejemplo (permitir HTTP): `sudo ufw allow 80/tcp`
    - Ejemplo (permitir HTTPS): `sudo ufw allow 443/tcp`
    - Ejemplo (permitir DNS UDP): `sudo ufw allow 53/udp`
    - Ejemplo (permitir tráfico en un rango de puertos): `sudo ufw allow 6000:6007/tcp`
- **`ufw deny <puerto>/<protocolo>`:** Deniega el tráfico a un puerto específico y protocolo.
    - Ejemplo (denegar tráfico a VNC): `sudo ufw deny 5900/tcp`
- **`ufw allow <nombre_servicio>`:** Permite el tráfico para servicios comunes definidos en `/etc/services`.
    - Ejemplo: `sudo ufw allow ssh` (equivalente a `allow 22/tcp`)
    - Ejemplo: `sudo ufw allow http` (equivalente a `allow 80/tcp`)
    - Ejemplo: `sudo ufw allow https` (equivalente a `allow 443/tcp`)
    - Ejemplo: `sudo ufw allow mysql` (equivalente a `allow 3306/tcp`)

Por Dirección IP:

- **`ufw allow from <IP>`:** Permite todo el tráfico desde una IP específica.
    - Ejemplo: `sudo ufw allow from 192.168.1.100`
- **`ufw deny from <IP>`:** Deniega todo el tráfico desde una IP específica.

    - Ejemplo: `sudo ufw deny from 192.168.1.101`
- **`ufw allow from <IP> to any port <puerto>`:** Permite tráfico desde una IP a un puerto específico.
    - Ejemplo: `sudo ufw allow from 192.168.1.100 to any port 22`
- **`ufw allow from <red>/<máscara> to any port <puerto>`:** Permite tráfico desde una subred a un puerto específico.
    - Ejemplo: `sudo ufw allow from 192.168.1.0/24 to any port 3306`

Por Interfaz de Red:

- **`ufw allow in on <interfaz> to any port <puerto>`:** Permite tráfico entrante en una interfaz específica a un puerto.
    - Ejemplo: `sudo ufw allow in on eth0 to any port 80`
- **`ufw allow out on <interfaz> to any port <puerto>`:** Permite tráfico saliente en una interfaz específica desde un puerto.
    - Ejemplo: `sudo ufw allow out on eth0 to any port 53`

Reglas de Salida (Outgoing Rules):

Aunque la política por defecto es `allow outgoing`, puedes definir reglas específicas para el tráfico saliente.

- **`ufw allow out to <IP> port <puerto>`:** Permite tráfico saliente a una IP y puerto específicos.
    - Ejemplo: `sudo ufw allow out to 8.8.8.8 port 53/udp` (permitir consultas DNS solo a Google DNS).
- **`ufw deny out to any port 25`:** Denegar todo el tráfico saliente al puerto 25 (SMTP) para prevenir spam no autorizado.

5. Eliminar Reglas

Puedes eliminar reglas por su número (mostrado con `ufw status numbered`) o por la regla explícita.

- **`ufw delete <número_de_regla>`:** Elimina una regla por su índice numérico.
    - Ejemplo (ver números): `sudo ufw status numbered`
    - Ejemplo (eliminar la regla número 3): `sudo ufw delete 3`
- **`ufw delete <regla_completa>`:** Elimina una regla especificando la regla exacta que se añadió.
    - Ejemplo: `sudo ufw delete allow 22/tcp`
    - Ejemplo: `sudo ufw delete deny from 192.168.1.101`

6. Insertar Reglas

Puedes insertar una regla en una posición específica de la lista de reglas.

- **`ufw insert <posición> allow <regla>`:** Inserta una regla en la posición deseada. Las reglas se procesan en orden.
    - Ejemplo (insertar una regla para permitir SSH desde una IP específica antes de cualquier otra): `sudo ufw insert 1 allow from 192.168.1.50 to any port 22`

7. Habilitar el Registro (Logging)

El registro (logging) es crucial para monitorear la actividad del firewall, detectar intentos de ataque y depurar problemas.

- **`ufw logging on`:** Habilita el registro del firewall. Los logs se enviarán a `/var/log/syslog` o `/var/log/messages` (dependiendo de la distribución) y también pueden ser vistos con `journalctl`.
    - Ejemplo: `sudo ufw logging on`
- **`ufw logging off`:** Deshabilita el registro.
    - Ejemplo: `sudo ufw logging off`
- **`ufw logging low|medium|high|full`:** Establece el nivel de verbosidad del registro. `low` es el por defecto. `full` es muy ruidoso y puede llenar rápidamente los logs.
    - Ejemplo: `sudo ufw logging high`

8. Perfiles de Aplicación (Application Profiles)

UFW puede usar perfiles predefinidos para aplicaciones comunes (si están instaladas y tienen un perfil UFW).

- **`ufw app list`:** Muestra una lista de perfiles de aplicación disponibles.
    - Ejemplo: `sudo ufw app list`
- **`ufw app info <nombre_perfil>`:** Muestra información detallada sobre un perfil de aplicación.
    - Ejemplo: `sudo ufw app info "OpenSSH"`
- **`ufw allow <nombre_perfil>`:** Permite el tráfico según las reglas definidas en el perfil de la aplicación.
    - Ejemplo: `sudo ufw allow "OpenSSH"`

9. Reglas Avanzadas y de Enrutamiento

UFW también soporta algunas reglas más avanzadas, aunque su fortaleza reside en la simplicidad para el tráfico unidireccional.

- **Reglas de enrutamiento:** Si tu servidor actúa como un router (reenviando tráfico entre interfaces), necesitas habilitar el reenvío IP en el kernel y luego usar reglas `ufw route`. Esto es más complejo y generalmente se gestiona editando `/etc/ufw/sysctl.conf` y `ufw before.rules`.

    - Ejemplo (permitir tráfico de reenvío en el puerto 80): `sudo ufw route allow in on eth0 out on eth1 to any port 80`
- **`ufw limit <puerto>/<protocolo>`:** Limita las conexiones a un puerto para prevenir ataques de fuerza bruta. Bloquea si una IP intenta iniciar 6 o más conexiones en 30 segundos.
    - Ejemplo: `sudo ufw limit 22/tcp` (muy recomendado para SSH).

10. Archivos de Configuración

Aunque UFW simplifica la gestión, las reglas finales se traducen a `iptables`. Los archivos de configuración de UFW están en `/etc/ufw/`.

- `/etc/default/ufw`: Configuración general de UFW (políticas por defecto).
- `/etc/ufw/before.rules` y `/etc/ufw/after.rules`: Permiten añadir reglas de `iptables` personalizadas que se aplican antes o después de las reglas de UFW. Útil para reglas `NAT` o `MASQUERADE`.
- `/etc/ufw/user.rules` y `/etc/ufw/user6.rules`: Contienen las reglas que añades con `ufw allow/deny`.

Flujo de Trabajo Recomendado para Configurar UFW:

1. **Conectarse vía SSH (si es un servidor remoto).**
2. **Permitir SSH antes de habilitar:** `sudo ufw allow 22/tcp`
3. **Establecer políticas por defecto:** `sudo ufw default deny incoming` `sudo ufw default allow outgoing`
4. **Habilitar el firewall:** `sudo ufw enable` (confirma con 'y').
5. **Verificar el estado:** `sudo ufw status verbose`
6. **Añadir reglas para otros servicios necesarios:**
    - Web (HTTP/S): `sudo ufw allow 80/tcp` `sudo ufw allow 443/tcp`
    - Base de datos (ej. MySQL): `sudo ufw allow from <IP_servidor_web> to any port 3306`
    - DNS: `sudo ufw allow 53` (si el servidor es un servidor DNS)
7. **Habilitar el registro (opcional, pero recomendado):** `sudo ufw logging on`
8. **Revisar los logs:** `tail -f /var/log/syslog | grep "UFW"` (o `journalctl -f | grep "UFW"`)

Consideraciones al Usar UFW

- **Orden de las reglas:** Las reglas se procesan en el orden en que aparecen. Una regla `deny` colocada antes de una `allow` para el mismo tráfico prevalecerá. `ufw status numbered` es crucial para ver el orden.
- **SSH Remoto: Siempre, siempre, siempre** permite SSH antes de habilitar el firewall si estás conectado remotamente. Un error puede dejarte fuera del servidor.

- **Recarga de reglas:** Cuando añades o eliminas reglas con `ufw`, estas se aplican inmediatamente. No necesitas reiniciar UFW.
- **Reinicios:** Las reglas de UFW son persistentes por defecto y se recargan automáticamente al reiniciar el sistema.
- **Complejidad:** Para configuraciones de firewall muy complejas (como VPNs con routing, NAT avanzado, balanceo de carga), `iptables` directo o `nftables` (el sucesor de `iptables`) pueden ser más adecuados, pero UFW cubre la gran mayoría de los casos de uso comunes.