M6 – P2: Configuración de Firewall en Linux y Windows

El firewall es una de las primeras líneas de defensa en seguridad de red. Su función es controlar el tráfico de red permitiendo o bloqueando paquetes según reglas establecidas. Esta práctica busca configurar firewalls básicos en **Linux,** usando UFW, y **Windows,** usando el firewall nativo, para reducir la superficie de ataque de los sistemas.

Objetivos específicos

- Comprender el rol del firewall como medida de hardening.
- Configurar reglas de firewall en entornos Linux y Windows.
- Identificar puertos abiertos y limitar el acceso solo a servicios necesarios.
- Validar la efectividad del firewall usando herramientas como `nmap`.

Requisitos técnicos

- Dos máquinas virtuales (1 Kali Linux, 1 Windows 10 o 11)
- Usuario con permisos administrativos (`sudo` en Linux)
- Conexión de red entre las VMs (adaptador puente o red interna)
- Herramienta `nmap` (ya incluida en Kali)

1. Configuración de Firewall en Linux: UFW Uncomplicated Firewall

1.1. Habilitar UFW

Actualizar e instalar UFW

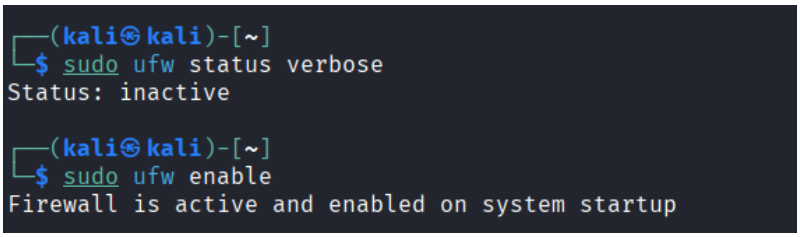
```
sudo apt update
sudo apt install ufw -y
```

Verificar estado

```
sudo ufw status verbose
```

Habilitar UFW

```
sudo ufw enable
```



Importante: Al habilitar UFW, asegúrate de no bloquear el acceso SSH si estás conectado remotamente:

```
sudo ufw allow ssh
```

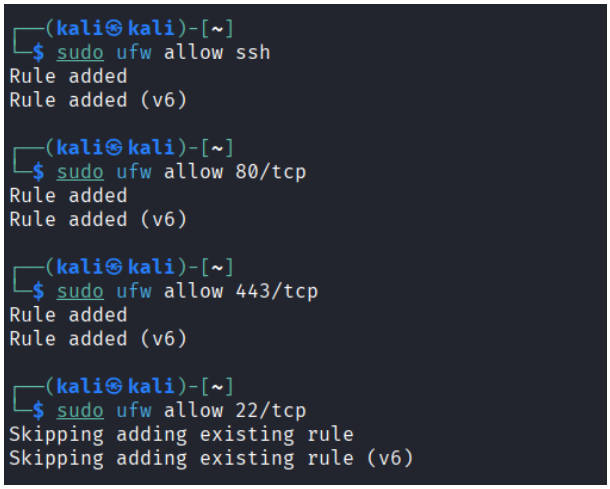
O alternativamente:

```
sudo ufw allow 22/tcp
```

1.2. Permitir puertos esenciales

Puertos básicos de servicios comunes

```
sudo ufw allow 22/tcp    # SSH - administración remota
sudo ufw allow 80/tcp    # HTTP - servidor web
sudo ufw allow 443/tcp   # HTTPS - servidor web seguro
```



Justificación de puertos permitidos:

- **Puerto 22 (SSH)**: Esencial para administración remota segura
- **Puerto 80 (HTTP)**: Necesario para servicios web
- **Puerto 443 (HTTPS)**: Para servicios web seguros
- **Otros puertos** se bloquean para reducir superficie de ataque

1.3. Establecer políticas por defecto

- Bloquear todo tráfico entrante no permitido

```
sudo ufw default deny incoming
```

- Permitir todo tráfico saliente

```
sudo ufw default allow outgoing
```

Esto bloquea todo lo que no se permita explícitamente, muy recomendado.

1.4. Agregar y eliminar reglas adicionales

- Permitir solo tráfico ICMP (ping) desde red local:

```
sudo ufw allow from 192.168.1.0/24 to any proto icmp
```

- Bloquear servicios inseguros

```
sudo ufw deny 23/tcp    # Telnet - sin cifrado
sudo ufw deny 21/tcp    # FTP - sin cifrado
sudo ufw deny 445/tcp   # SMB - potencial vulnerabilidad
```

- Eliminar reglas

```
sudo ufw delete allow 23/tcp
```

- Ver reglas numeradas para eliminación

```
sudo ufw status numbered
```

1.5. Verificar configuración

- Ver configuración completa

```
sudo ufw status verbose
sudo ufw status numbered
```

1.6. Prueba técnica: Desde Kali (u otra VM), ejecutar:

```
nmap -sS -p- 10.0.0.130
nmap -sS -p 22,80,443,23,21 10.0.0.130
```

2. Configuración de Firewall en Windows 10/11

2.1. Acceder a las opciones del Firewall

Método 1 - Panel de Control:

1. Panel de Control - Sistema y Seguridad - Firewall de Windows Defender
2. Configuración avanzada

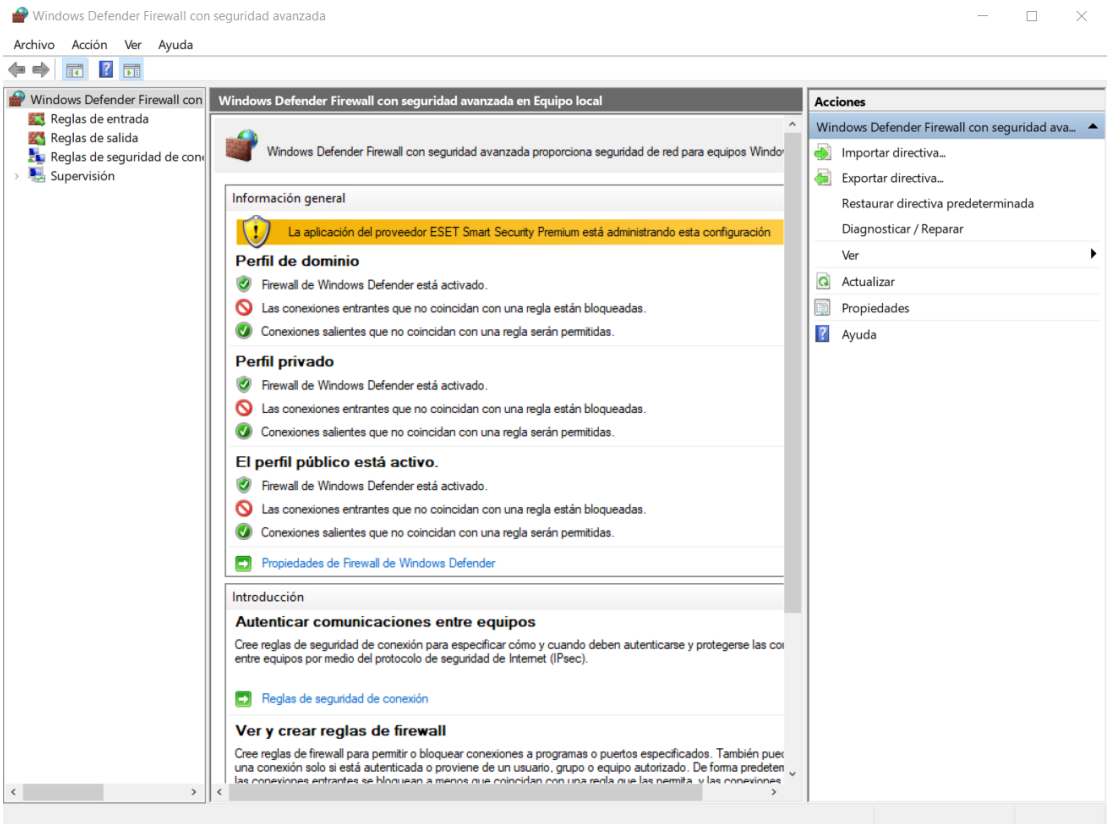
Método 2 - Comandos rápidos:

Abrir firewall directamente

```
wf.msc
```

Familiarizarse con las categorías:

1. Reglas de entrada
2. Reglas de salida
3. Perfiles de red



2.2. Crear nuevas reglas

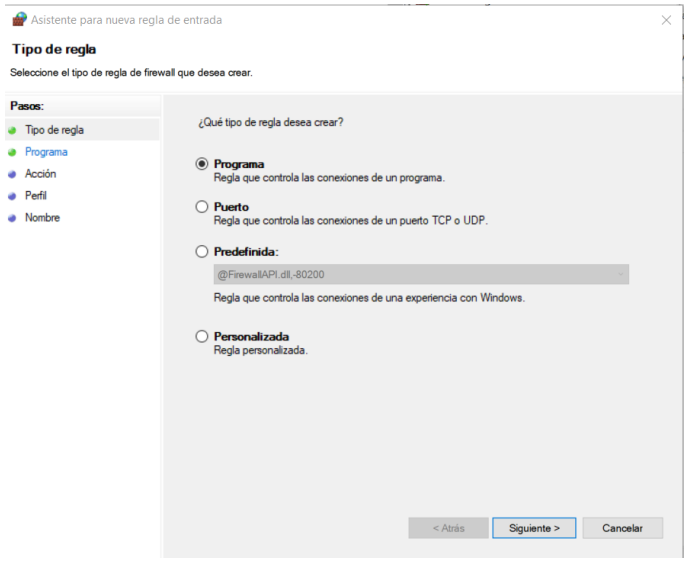
Regla 1 - Permitir RDP desde IP específica

Comando PowerShell equivalente

```
New-NetFirewallRule -DisplayName "RDP desde IP específica" `
-Direction Inbound -Protocol TCP -LocalPort 3389 `
-Action Allow -RemoteAddress 192.168.1.10
```

Configuración gráfica:

- Tipo: Personalizada
- Programa: Todos los programas
- Protocolo: TCP
- Puerto local: 3389
- Dirección remota: 192.168.1.10
- Acción: Permitir la conexión



Regla 2 - Bloquear SMB (puerto 445)

```
New-NetFirewallRule -DisplayName "Bloquear SMB" `
-Direction Inbound -Protocol TCP -LocalPort 445 `
-Action Block
```

Regla 3 - Permitir HTTP para todos

```
New-NetFirewallRule -DisplayName "Permitir HTTP" `
-Direction Inbound -Protocol TCP -LocalPort 80 `
-Action Allow
```

Muy importante es asignar nombres claros a las reglas.

2.3. Verificar reglas

Desde Kali o la segunda VM ejecutar los siguientes comandos desde el bash:

- Escanear puertos específicos

  ```
  nmap -p 445,3389,80 IP_Windows
  ```

- Escaneo completo

  ```
  nmap -sS -T4 IP_Windows
  ```

Validar si las reglas funcionan como se espera según la configuración establecida.

2.4. Auditar tráfico bloqueado

Habilitar registro de eventos bloqueados:

1. Firewall con seguridad avanzada
2. Propiedades del firewall (para cada perfil)
3. Configurar: "Sí" en "Registrar paquetes descartados"
4. Mostrar los logs,
   - Ruta del log

     ```
     C:\Windows\System32\LogFiles\Firewall\pfirewall.log
     ```

   - Usar PowerShell para ver logs recientes

     ```
     Get-Content "C:\Windows\System32\LogFiles\Firewall\pfirewall.log" -Tail 50
     ```
5. Simular un escaneo desde otra máquina.
6. Analizar el log y documentar una línea de tráfico bloqueado.

3. Actividades complementarias

- Simular un ataque por fuerza bruta (Hydra, Nmap scripts) y demostrar cómo un firewall puede bloquearlo.
- Hacer ping y escaneos desde diferentes subredes para ver cómo se comporta el firewall.

3.1. Simulación de Ataque y Protección

Desde Kali Linux:

- Escaneo de vulnerabilidades SMB

```
nmap --script smb-vuln* -p 445 192.168.1.50
```

- Fuerza bruta SSH, solo en entorno controlado

```
hydra -L usuarios.txt -P claves.txt ssh://192.168.1.100
```

3.2. Pruebas de Conectividad

- Verificar bloqueo de ping

```
ping 192.168.1.100
```

- Escanear desde diferentes subredes

```
nmap -p 22,80,443 192.168.1.100
```

- Verificar servicios específicos

```
telnet 192.168.1.100 23  # Debe fallar
```

3.3. Análisis de Resultados

Antes de Configurar Firewall:

Ejemplo de escaneo inicial

```
nmap -sS 192.168.1.100
```

Resultado típico: Múltiples puertos abiertos

Después de Configurar Firewall:

Escaneo posterior a configuración

```
nmap -sS 192.168.1.100
```

Resultado esperado: Solo puertos 22, 80, 443 abiertos

4. Diferencias entre Windows y Linux:

Windows Firewall:

- Interfaz gráfica más intuitiva
- Integración nativa con el sistema
- Perfiles separados (dominio, privada, pública)

Linux UFW:

- Configuración por línea de comandos
- Mayor flexibilidad para scripts
- Integración con iptables

5. Impacto de Permitir Todos los Puertos:

- **Aumento exponencial** de superficie de ataque
- Servicios vulnerables expuestos
- Mayor probabilidad de compromiso

6. Servicios que Deben Estar Siempre Bloqueados:

- Telnet (23) - Sin cifrado
- FTP (21) - Sin cifrado
- SMB (445) - Históricamente vulnerable
- RPC (135) - Múltiples vulnerabilidades
- NetBIOS (137-139) - Obsoleto e inseguro

7. Mejores Prácticas Documentadas

Para Linux (UFW):

En en bash de Linux ejecutar:

1. Siempre permitir SSH primero

```
sudo ufw allow ssh
```

2. Políticas restrictivas por defecto

```
sudo ufw default deny incoming

sudo ufw default allow outgoing
```

3. Solo servicios necesarios

```
sudo ufw allow 22/tcp

sudo ufw allow 80/tcp

sudo ufw allow 443/tcp
```

4. Bloquear servicios inseguros

```
sudo ufw deny 23/tcp

sudo ufw deny 21/tcp
```

5. Verificar configuración

```
sudo ufw status verbose
```

Para Windows Firewall:

En PowerShell de Windows ejecutar:

1. Habilitar firewall para todos los perfiles

```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
```

2. Configurar políticas por defecto

```
Set-NetFirewallProfile -DefaultInboundAction Block -DefaultOutboundAction All
ow
```

3. Crear reglas específicas: Permitir RDP desde IP específica

```
New-NetFirewallRule -DisplayName "RDP Controlado" -Direction Inbound -Protoco
l TCP -LocalPort 3389 -Action Allow -RemoteAddress 192.168.1.0/24
```

4. Bloquear servicios riesgosos

```
New-NetFirewallRule -DisplayName "Bloquear SMB" -Direction Inbound -Protocol
TCP -LocalPort 445 -Action Block
```

8. Validación Final

Checklist de Seguridad:

- Solo puertos necesarios están abiertos
- Servicios inseguros bloqueados
- Políticas por defecto son restrictivas

- Logs de firewall habilitados
- Reglas documentadas y justificadas
- Comunicación esencial funciona
- Accesos no autorizados bloqueados

Comandos de Verificación Final:

Desde Kali Linux - Verificar configuración

```
nmap -sS -T4 -p- 192.168.1.100

nmap -sS -T4 -p- 192.168.1.50
```

Verificar servicios específicos

```
telnet 192.168.1.100 22    # Debe conectar con el objetivo

telnet 192.168.1.100 23    # Debe fallar la conexión
```