**Laboratorio IPTABLES:**

¿Qué es iptables y para qué sirve?:

Iptables es una herramienta esencial en Linux, diseñada para gestionar y filtrar el tráfico de red a través de reglas definidas. Su nombre, derivado de "IP" y "tables", refleja su función principal de trabajar con tablas que contienen reglas específicas para determinar el destino de los paquetes de datos. Este sistema de filtrado de paquetes ofrece varias funcionalidades, desde las tareas más básicas hasta funciones avanzadas de administración de red.

Con iptables, es posible bloquear o permitir el tráfico basándose en criterios como direcciones IP, puertos, protocolos y estados de conexión. Además, una de las características más destacadas de iptables es su capacidad para realizar la traducción de direcciones de red, permitiendo compartir una única dirección IP pública entre varios dispositivos en una red interna. También puede realizar mapeo de puertos, redirigiendo el tráfico desde un puerto específico a otro, facilitando la exposición de servicios internos a través de una única dirección IP pública.

Iptables mantiene un seguimiento del estado de las conexiones, recordando si una conexión está establecida, en curso o es una respuesta a una conexión interna. Esto proporciona una capacidad avanzada para establecer reglas de estado y gestionar el tráfico de manera eficiente.

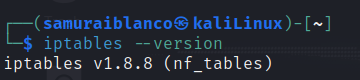
En términos de configuración, iptables utiliza reglas organizadas en diferentes tablas, como la tabla "filter" para el filtrado básico, la tabla "nat" para la traducción de direcciones, y la tabla "mangle" para la manipulación de paquetes.

Web: <https://www.netfilter.org/projects/iptables/index.html>

**DESARROLLO**

**Verificar si iptables está instalado en el sistema:**

*iptables --version*

****

En caso de no estar instalado, en Debian o sus derivados, seguir los siguientes pasos para instalar:

*Desinstala nftables (si está instalado): Primero, desinstala nftables y sus dependencias.*

*sudo apt-get remove --auto-remove nftables*

*sudo apt-get purge nftables*

*Instala iptables: A continuación, instala iptables ejecutando los siguientes comandos. Esto descargará e instalará iptables en tu sistema:*

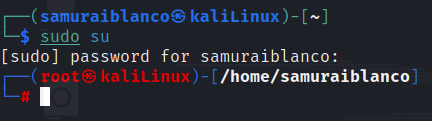
*sudo apt-get update*

*sudo apt-get install iptables*

*Verifica el estado de iptables: Para asegurarte de que iptables se haya instalado correctamente, puedes ejecutar:*

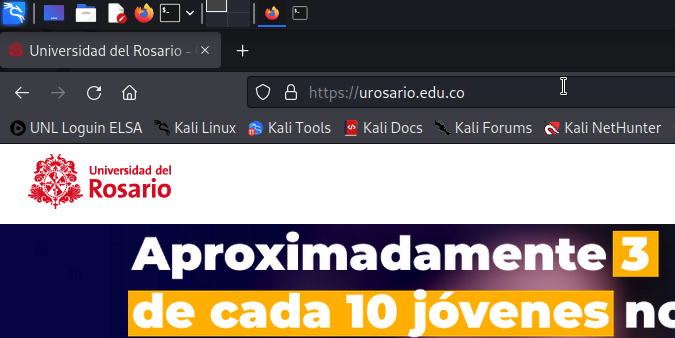
*sudo iptables --version*

**Para trabajar más “cómodamente”, se sugiere pasar a modo root:**



El presente laboratorio se llevará a cabo en Kali Linux y como pudimos apreciar anteriormente, iptables ya se encuentra instalado por defecto.

Para verificar si tenemos salida a internet, consultemos desde Firefox la página de la universidad:



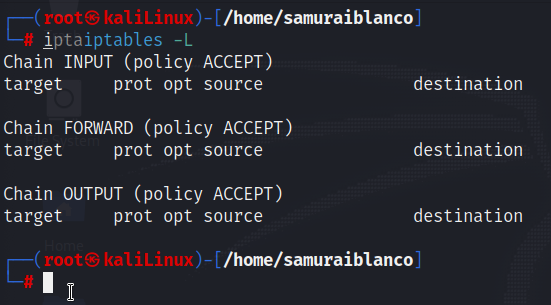
Efectivamente hay salida a internet.

Lo primero que podemos hacer es borrar todas las reglas del firewall para aquellos casos donde hayamos cometido errores o queramos empezar desde cero:

*iptables -F*

**Verificar si las tablas quedaron en blanco**

*iptables -L*

****

Efectivamente no hay registros en las tablas

**Denegar o aceptar el tráfico**

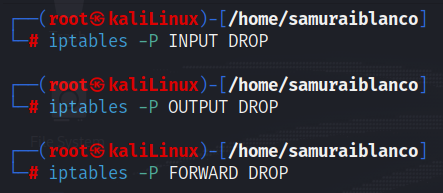
Podemos denegar todo el tráfico y luego ir añadiendo reglas que permitan el tráfico que queramos, o bien, podemos permitir todo el tráfico y luego añadir reglas que rechacen el que queramos.

Para ser óptimos en la configuración, vamos a elegir rechazar todo el tráfico inicialmente:

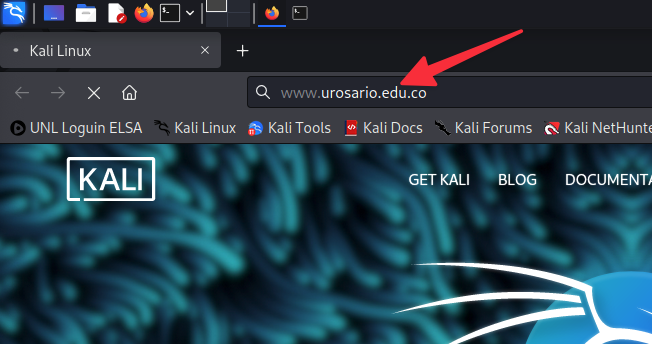
*iptables -P INPUT DROP*

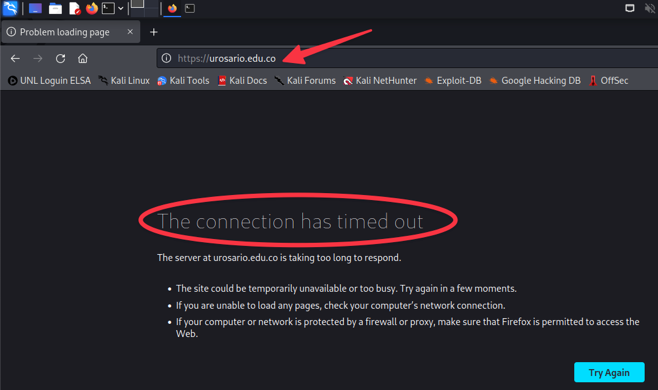
*iptables -P OUTPUT DROP*

*iptables -P FORWARD DROP*



Para verificar si efectivamente no hay navegación, intenta ingresar por Firefox nuevamente a la página de la Universidad. Podrás darte cuenta que el browser trata de procesar la petición pero la página web de UROSARIO no se despliega porque el firewall lo está bloqueando:





Listemos todas las reglas del firewall con este comando:

*iptables -L -nv --line-numbers*

**iptables:** Es el comando principal para administrar las reglas del firewall en sistemas Linux. Puedes usarlo para configurar, mostrar y manipular las reglas de filtrado de paquetes.

**-L:** Esta opción indica que queremos listar las reglas. En este caso, estamos solicitando una lista completa de todas las reglas.

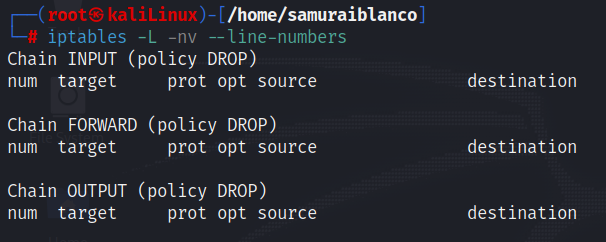
-**nv:** Estas opciones adicionales tienen los siguientes significados:

-**n:** Muestra las direcciones IP y puertos en formato numérico en lugar de resolver nombres de host y servicios.

-**v:** Proporciona una salida más detallada, incluyendo información sobre paquetes y bytes afectados por cada regla.

**--line-numbers:** Esta opción agrega números de línea a la salida, lo que facilita la referencia a reglas específicas.

En resumen, el comando iptables -L -nv --line-numbers mostrará todas las reglas del firewall iptables en tu sistema, incluyendo detalles como las direcciones IP, puertos y estadísticas asociadas con cada regla.



Y ahora si queremos permitir el tráfico del firewall, ejecutamos estos otros comandos:

*iptables -P INPUT ACCEPT*

*iptables -P OUTPUT ACCEPT*

*iptables -P FORWARD ACCEPT*

Verifica de nuevo por Firefox ingresar a la página de la universidad y podrás darte cuenta que nuevamente tenemos acceso.

**SCRIPT PARA ELIMINAR TRÁFICO E IR PERMITIENDO POCO A POCO SEGÚN LA NECESIDAD**

Creamos un script que en principio se encarga de denegar todo el tráfico y luego en dicho script se va contemplando aquel tráfico que queramos permitir:

*nano ipt-denegar.sh*

#!/bin/bash

#Eliminar reglas en las tablas filter y nat:

iptables -t filter -F

iptables -t nat -F

# Reiniciar los contadores en las tablas filter y nat:

iptables -t filter -Z

iptables -t nat -Z

# Denegar todo el tráfico de las tres cadenas de filter (INPUT, OUTPUT, FORWARD)

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

Después de guardar cambios asignamos permisos al script creado (*Se sugiere por seguridad otorgarle todos los permisos SOLO al usuario que ejecutará el script. Por temas de facilidad para el laboratorio, en el siguiente comando se le otorgaran todos los permisos a todos los usuarios; en todo caso, tendrás que ejecutarlo con root*):

*chmod 777 ipt-denegar.sh*

Posteriormente ejecutamos el script:

*./ipt-denegar.sh*

Intentamos nuevamente en Firefox ingresar a la web de la universidad y podremos darnos cuenta que no ha sido posible. Por lo cual, el script está funcionando correctamente.

**Nota: si quisiéramos crear un script para permitir todo el tráfico, podríamos hacerlo cambiando en el script anterior la palabra “DROP” por la palabra “ACCEPT”.**

**IR AÑADIENDO REGLAS PARA PERMITIR CIERTO TRÁFICO**

**(desde script restrictivo que limita todo el tráfico)**

Tenemos activado el script que deniega todo el tráfico, y ahora podemos ir permitiendo regla a regla; por ejemplo vamos a permitir que haya tráfico ICMP tanto de salida como de entrada agregando las siguientes líneas al final del script:

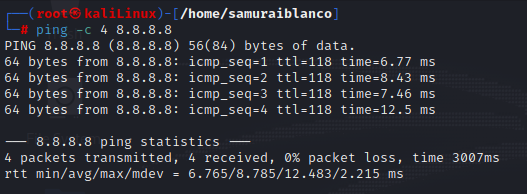
# Añadimos reglas para permitir el ping de entrada y de salida (tráfico ICMP)

iptables -A OUTPUT -o eth0 -p icmp -j ACCEPT

iptables -A INPUT -i eth0 -p icmp -j ACCEPT

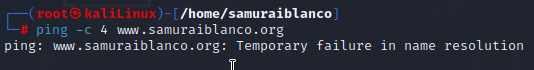
Realizamos prueba de ping a uno de los DNS de Google:

*ping -c 4 8.8.8.8*



Lo que pasa con las líneas agregadas anteriormente al script es que sólo hemos permitido el tráfico de ping, pero nos falta el tráfico DNS, ya que ahora mismo si hacemos un ping a un dominio, no se podrá resolver:

*ping -c 4 www.samuraiblanco.org*



Para permitir el tráfico DNS agregamos las siguientes líneas al final del script donde además también habilitamos la interfaz loopback:

# Consultas y respuestas DNS

iptables -A OUTPUT -o eth0 -p udp --dport 53 -j ACCEPT

iptables -A INPUT -i eth0 -p udp --sport 53 -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp --dport 53 -j ACCEPT

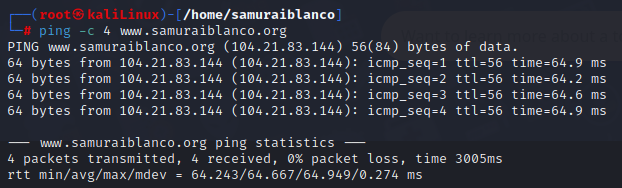
iptables -A INPUT -i eth0 -p tcp --sport 53 -j ACCEPT

# Habilitar interfaz loopback en systemd-resolve

iptables -I INPUT 1 -i lo -j ACCEPT

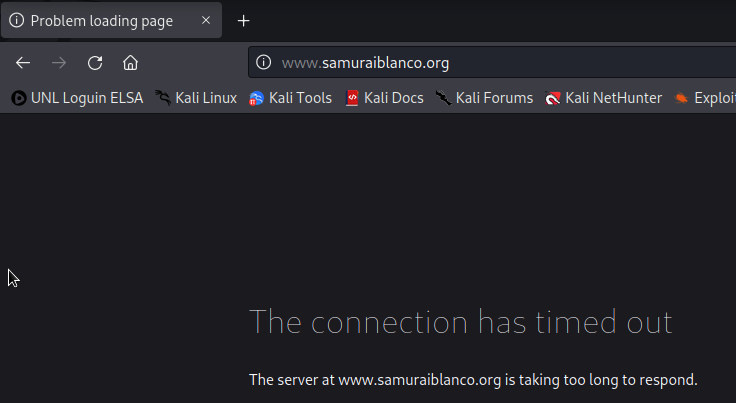
iptables -I OUTPUT 1 -o lo -j ACCEPT

Ahora que hemos agregado/permitido el tráfico DNS, volvemos a realizar un ping al dominio www.samuraiblanco.org y podremos verificar que efectivamente se encuentra respondiendo este tráfico (DNS):



**REGULAR TRÁFICO HTTP Y HTTPS NAVEGADOR WEB**

En este punto, no podemos navegar por la web (en ambiente gráfico), ya que el firewall está bloqueando este tipo de tráfico, por lo que tendremos que añadir nuevas reglas para regularlo, por ejemplo vamos a habilitar el tráfico por el puerto 80:



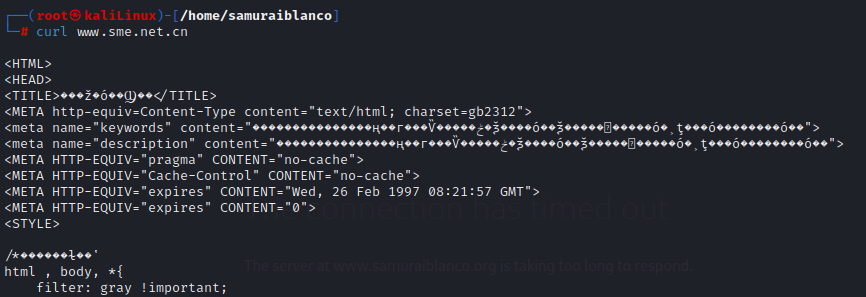
Agregamos las siguientes reglas al final del script:

# Habilitar trafico HTTP – Puerto 80

iptables -A OUTPUT -o eth0 -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --sport 80 -j ACCEPT

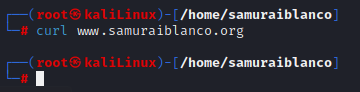
Al hacer un curl a la página www.sme.net.cn vemos que hay respuesta:



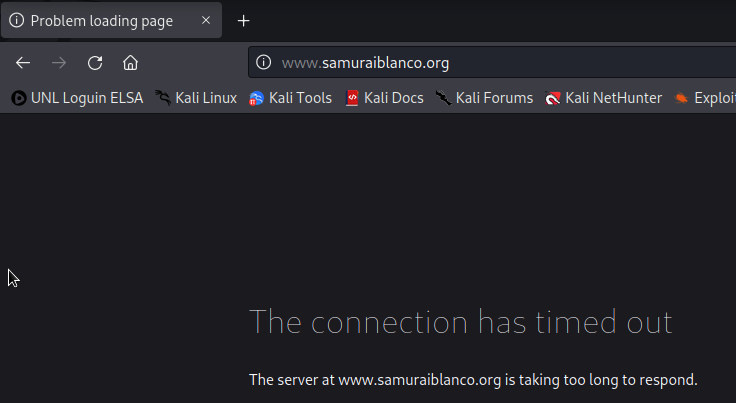
Así mismo, al consultar www.sme.net.cn por Firefox podemos apreciar que se despliega la web:



Sin embargo, al hacer un curl al dominio www.samuraiblanco.org podemos ver que no hay ninguna respuesta:



Igualmente al consultar por Firefox el dominio www.samuraiblanco.org nuevamente es imposible poder desplegar la página:



El problema de no poderse desplegar www.samuraiblanco.org radica en que está página se está sirviendo por por https, por lo que tenemos que habilitar el tráfico https y este se mueve por el puerto 443; para ello deberemos habilitar el tráfico también por este puerto agregando las siguientes líneas al script:

# Habilitar trafico HTTPS – Puerto 443

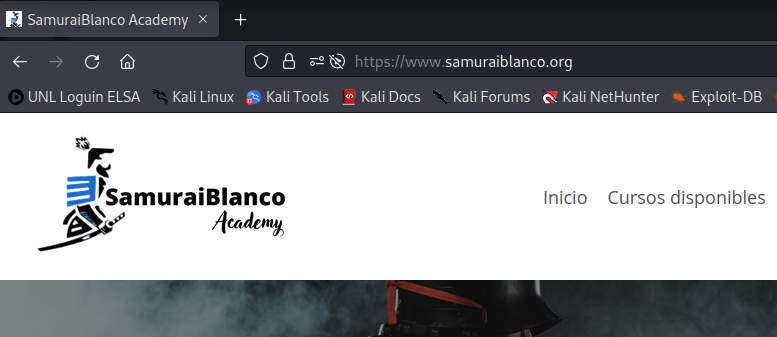
iptables -A OUTPUT -o eth0 -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --sport 443 -j ACCEPT

Realizando nuevamente una comprobación con curl al dominio https://www.samuraiblanco.org (fíjate que le indicamos el protocolo es https) y ya obtenemos respuesta:



Y al verificar por Firefox el dominio www.samuraiblanco.org ya obtenemos respuesta de la página:

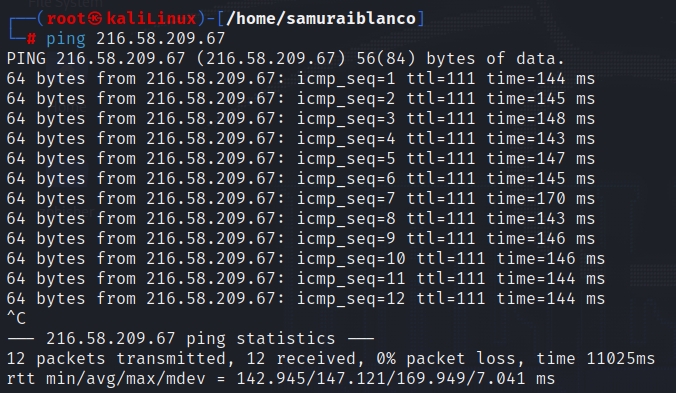


**NOTA:** Realiza la prueba de reiniciar la máquina y verás que las reglas creadas no son persistentes. Puedes verificar tráfico, ping, ver tablas, etc. No te preocupes, más adelante veremos como configurar las reglas de manera persistente.

**BLOQUEAR DIRECCIÓN IP EN FIREWALL IPTABLES**

Realiza el siguiente ping para probar la respuesta:

*ping a 216.58.209.67*



**Pequeñas pruebas de concepto**

**Primero:** Para realizar el bloqueo de ip agregamos las siguientes líneas al final del script:

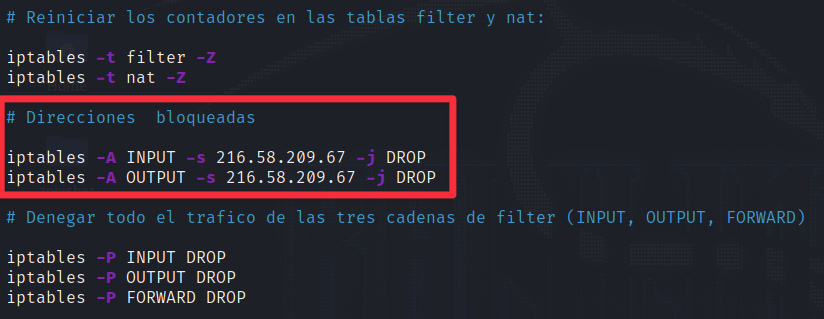
# Direcciones bloqueadas

iptables -A INPUT -s 216.58.209.67 -j DROP

iptables -A OUTPUT -s 216.58.209.67 -j DROP

Ejecutamos el script y posteriormente vuelve a hacer el ping anterior. Qué conclusión sacas?

**Segundo:** Elimina las líneas de la parte final del script y colócalas entre las siguientes líneas así:



Realiza nuevamente el ping después de ejecutar el script y saca tu propia conclusión.

**Tercero:** Cambia las líneas anteriores a diferentes partes del script y comprueba los resultados del ping después de cada cambio.

**Otra opción para el bloqueo del ejemplo anterior**

También podemos limitar sólo el tráfico por ICMP de la IP (no es necesario que lo compruebes en el script):

# Limitar sólo el tráfico por ICMP de una IP

iptables -A OUTPUT -s 216.58.209.67 -p icmp -j ACCEPT

iptables -A INPUT -s 216.58.209.67 -p icmp -j ACCEPT

**INFORMACIÓN COMPLEMENTARIA**

**Para bloquear todo el tráfico de un puerto**

Por ejemplo, bloquear el tráfico del puerto 22 (ssh)

*iptables -A INPUT -p tcp --dport 22 -j DROP*

También podrías habilitar el tráfico por ssh sólo a una determinada dirección IP; y para ello lo harías de esta forma:

*iptables -A INPUT -p tcp --dport 22 -s 192.168.0.6 -j ACCEPT*

**Para hacer las reglas de iptables persistentes y asegurarte de que se mantengan después de un reinicio**

Puedes seguir una de las siguientes opciones:

1. **Usando iptables-persistent**

Instala el paquete iptables-persistent utilizando el administrador de paquetes apt

*apt install iptables-persistent*

Las reglas de iptables actualmente configuradas se guardarán en los siguientes archivos correspondientes a IPv4 e IPv6:

*/etc/iptables/rules.v4*

*/etc/iptables/rules.v6*

Estos archivos se cargarán automáticamente durante el arranque para restaurar las reglas.

1. **Guardando manualmente las reglas**

Si prefieres no usar iptables-persistent, puedes guardar manualmente tus reglas en un archivo y cargarlas después del reinicio.

Guarda tus reglas en un archivo (por ejemplo, my\_iptables\_rules.rules):

*iptables-save > /ruta/a/mi\_archivo\_de\_reglas.rules*

Luego, para cargar las reglas desde el archivo:

*iptables-restore < /ruta/a/mi\_archivo\_de\_reglas.rules*

1. **Automatización con scripts de inicio**

Puedes crear un script de inicio personalizado (por ejemplo, en /etc/rc.local) que cargue tus reglas de iptables al arrancar el sistema.

**Para eliminar una regla en iptables**

Sigue los siguientes pasos

**Listar las reglas existentes:** primero, debes identificar la regla que deseas eliminar. Puedes listar todas las reglas de iptables para ver cuál deseas borrar. Ejecuta el siguiente comando (hay varias formas para ello pero esta es la que te recomiendo porque así puedes ver el ID de cada regla):

*iptables -L -nv --line-numbers*

**Identificar la regla a eliminar**

Una vez que sepas qué regla deseas eliminar, toma nota del **número de cadena (ID)** y **línea de la regla**.

**Eliminar la regla**

Ejecuta el comando **iptables -D** seguido del número de cadena y línea de la regla.

Por ejemplo, esto eliminará la tercera regla de la cadena ENTRADA:

*iptables -D INPUT 3*

El uso de iptables es más amplio y cuenta con más opciones. Te sugiero ingresar a la web oficial y profundizar más en el tema:

[*https://iptables.org/projects/iptables/index.html*](https://iptables.org/projects/iptables/index.html)

Ahora, ingresa al aula virtual de la materia Seguridad en Redes y desarrolla el laboratorio/tarea para entregar.