

Proyecto 2

Nombre de la materia: Intercomunicación y seguridad en redes

Alumno: Brayan Blancas Monsalvo

Matricula: 201965973

Docente: Josué Pérez Romero

Introducción

Este documento tiene como objetivo presentar el segundo proyecto de la materia "Intercomunicación y seguridad en redes". La idea central del trabajo fue desarrollar como primera actividad un programa que simule el funcionamiento básico de un firewall, que es como un guardia de seguridad para una red, decidiendo qué tráfico puede pasar y qué tráfico debe ser bloqueado. Para la segunda actividad,

En las siguientes secciones, primero vamos a explicar paso a paso cómo está estructurado este reporte para que sea fácil de seguir. Luego, en el desarrollo, nos adentraremos en el código del programa. Ahí describiremos los conceptos clave que utilizamos, como las "reglas" que definen si se permite o se deniega un paquete de datos, y cómo funciona cada parte del proceso para tomar esa decisión, desde que añadimos una regla hasta que el firewall decide el destino de un paquete simulado. Básicamente, te contaremos todo el "cómo se hizo" de una manera clara y directa.

Desarrollo

1. Simulación de Filtrado de paquetes (Firewall 7.4.1)

- Escribir un programa que simule un *firewall* de capa de red. Define una lista de reglas (ACL):
 - Permitir tráfico desde IP X
 - Denegar tráfico al puerto Y
 - Regla por defecto: Denegar todo
- El programa debe simular la llegada de un paquete (con IP de origen, IP de destino y puerto) y determinar si se permite o deniega.

Programa

La clase principal Firewall cuenta con una lista de reglas (initialmente vacía) además de dos funciones: add_rule y process_packet.

```
class Firewall:  
    def __init__(self):  
  
        def add_rule(self, value, rule_type, action="DENY"):
```

```
def _matches_rule(self, rule, src_ip, dst_ip, dst_port):
```

add_rule

La función `add_rule` permite insertar reglas al firewall indicando el valor (ya sea de la dirección IP o puerto), el tipo de regla (de tipo `src_ip` o `dst_port`) y la acción a realizar (DENY o ALLOW).

```
def add_rule(self, value, rule_type, action="DENY"):
    if(self.rules.get(value) != None):
        print("Esa regla ya existe!")

    self.rules.update({value : [rule_type, action] })
```

process_packet

La función `process_packet` permite determinar si un paquete es denegado o procesado. Recibe una dirección IP de origen, destino y puerto. Internamente retorna la acción a tomar (True para permitir o False para denegar) según una regla establecida. En caso de no encontrar una regla (`self.rules.get(value) == None`) retorna False.

```
def process_packet(self, src_ip, dst_ip, dst_port):
    print(f"Procesando paquete: {src_ip} -> {dst_ip}:{dst_port}")

    if (self.rules.get(src_ip) != None):
        print(f"Regla aplicada: valor={src_ip}, tipo de regla= {self.rules.get(src_ip)[0]}, accion= {self.rules.get(src_ip)[1]}")

        return True if self.rules.get(src_ip)[1] == "ALLOW" else False
    elif (self.rules.get(dst_port) != None):
        print(f"Regla aplicada: valor={dst_port}, tipo de regla= {self.rules.get(dst_port)[0]}, accion= {self.rules.get(dst_port)[1]}")

        return True if self.rules.get(dst_port)[1] == "ALLOW" else False

    return False
```

Funcionamiento

Se indican una serie de reglas al firewall.

```
firewall = Firewall()

firewall.add_rule('192.168.1.100', 'src_ip', 'ALLOW')
firewall.add_rule('22', "dst_port", 'ALLOW')
```

```
firewall.add_rule('443', "dst_port", 'ALLOW')
firewall.add_rule('10.0.0.5', 'src_ip', 'DENY')
```

Finalmente, se observa el resultado del procesamiento de los paquetes.

```
Procesando paquete: 192.168.1.100 -> 192.168.1.1:8080
Regla aplicada: valor=192.168.1.100, tipo de regla= src_ip, accion= ALLOW
Resultado: PERMITIDO
```

```
Procesando paquete: 192.168.1.50 -> 192.168.1.1:443
Regla aplicada: valor=443, tipo de regla= dst_port, accion= ALLOW
Resultado: PERMITIDO
```

```
Procesando paquete: 10.0.0.5 -> 192.168.1.1:80
Regla aplicada: valor=10.0.0.5, tipo de regla= src_ip, accion= DENY
Resultado: DENEGADO
```

```
Procesando paquete: 192.168.1.50 -> 192.168.1.1:22
Regla aplicada: valor=22, tipo de regla= dst_port, accion= ALLOW
Resultado: PERMITIDO
```

Si se procesa un paquete con dirección IP o puerto sin regla establecida, se denega el paquete.

```
result5 = firewall.process_packet("0.0.0.1", "132.145.78.1", "444")`
```

```
Procesando paquete: 0.0.0.1 -> 132.145.78.1:444
Resultado: DENEGADO
```

Conclusión
