Trabalho 3:
Implementação de um Firewall de Rede

Eduardo Marques – 211021004
Yan Tavares – 202014323

UnB – CIC0201 – Segurança Computacional
2025/1 – Profª Priscila Solis

1 Introdução

A segurança de redes é um pilar fundamental da tecnologia da informação. No cenário digital atual, a proteção de perímetros de rede contra acessos não autorizados é essencial para a integridade e confidencialidade dos dados. O firewall de rede é a principal ferramenta para essa proteção, atuando como um filtro de pacotes que inspeciona e controla o tráfego com base em um conjunto de regras de segurança.

Este trabalho detalha a implementação de um firewall stateful (que analisa o estado das conexões) em um ambiente de rede simulado. O objetivo foi segmentar a rede em zonas de segurança, uma rede pública (Internet), uma Zona Desmilitarizada (DMZ) e uma rede interna e aplicar políticas de segurança rigorosas usando a ferramenta `iptables` do Linux. A DMZ foi projetada para hospedar servidores publicamente acessíveis (como um servidor web), isolando-os da rede interna, que contém servidores críticos (como um servidor DHCP) e as estações de trabalho dos usuários.

1.1 Repositório

O repositório remoto contendo registros da implementação pode ser encontrado em `https://github.com/yantavares/segcomp`.

2 Metodologia e Ferramentas

2.1 Ferramentas Utilizadas

A simulação foi realizada utilizando software de código aberto para virtualização e emulação de redes, permitindo a criação de uma topologia complexa e funcional.

- **GNS3 (Graphical Network Simulator-3):** Plataforma principal para a criação da topologia, conexão dos dispositivos virtuais e captura de pacotes para análise.
- **VirtualBox:** Software de virtualização utilizado para criar a máquina virtual base que serviu de template para todos os nós da rede (roteadores, servidores e clientes).
- **Ubuntu Server 24.04 LTS:** Sistema operacional Linux escolhido para todas as VMs, por sua estabilidade e por conter as ferramentas necessárias como `Netplan`, `iptables`, e os daemons para os serviços de DHCP.

2.2 Topologia da Rede

A topologia implementada, conforme a Figura 1, consiste em cinco sub-redes interconectadas por dois roteadores. O `Router1` funciona como o firewall de borda, controlando todo o tráfego entre a Internet, a DMZ e as redes internas. A utilização de uma máquina virtual `Linux Server` como roteador foi uma escolha metodológica focada no aprendizado prático e na construção da solução de segurança do zero. Diferentemente de appliances de rede proprietários que abstraem detalhes, esta abordagem permitiu controle sobre a configuração e acesso a logs detalhados do sistema. Isso foi essencial para diagnosticar o comportamento da rede e aprofundar a compreensão dos mecanismos de firewall no ecossistema Linux.

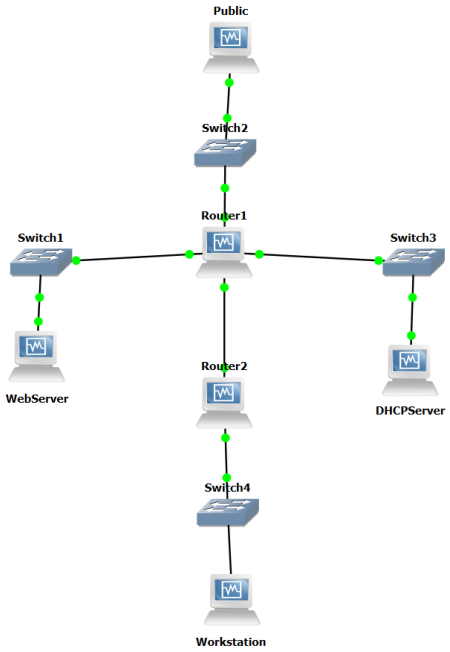

Figura 1: Topologia da rede implementada no GNS3.

3 Implementação e Configuração

A implementação seguiu uma abordagem metódica, iniciando pela configuração da camada de rede e roteamento, seguida pela configuração dos serviços e, finalmente, pela aplicação das regras do firewall.

3.1 Endereçamento IP e Configuração de Rede (Netplan)

A configuração de rede de todos os dispositivos foi tornada permanente utilizando o `Netplan`, o utilitário padrão do Ubuntu para configuração de redes. A Tabela 1 resume o plano de endereçamento.

Tabela 1: Plano de Endereçamento IP da Rede.

Dispositivo	**Interface (Linux)**	**Endereço IP**
Router1	enp0s9 (Internet)	172.16.0.254/24
	enp0s8 (DMZ)	10.0.20.254/24
	enp0s10 (Srv. Internos)	10.0.30.254/24
	enp0s16 (Link R2)	192.168.0.1/30
Router2	enp0s16 (Link R1)	192.168.0.2/30
	enp0s8 (Workstations)	10.0.40.254/24
WebServer	enp0s3	10.0.20.1/24
DHCPServer	enp0s3	10.0.30.1/24
WebTermPublic	enp0s3	172.16.0.1/24
WebTermWorkstation	enp0s3	(Via DHCP)

A configuração do Netplan no `Router1` é apresentada abaixo como exemplo.

Listing 1: Arquivo de configuração Netplan do Router1.

```
network:
  version: 2
  ethernets:
    enp0s8:
      addresses: [10.0.20.254/24]
    enp0s9:
      addresses: [172.16.0.254/24]
    enp0s10:
```

```
      addresses: [10.0.30.254/24]
    enp0s16:
      addresses: [192.168.0.1/30]
      routes:
        - to: 10.0.40.0/24
          via: 192.168.0.2
```

3.2 Configuração dos Serviços (DHCP e Relay)

O serviço de DHCP foi implementado utilizando o pacote `isc-dhcp-server` no `DHCPServer`. A configuração em `/etc/dhcp/dhcpd.conf` foi ajustada para incluir a diretiva `authoritative;` e uma `shared-network`, resolvendo um problema de interpretação de pedidos retransmitidos.

No `Router2`, o serviço de relay foi implementado com o pacote `isc-dhcp-relay`, para garantir o encaminhamento correto dos pacotes `DHCPDISCOVER`.

3.3 Implementação do Firewall (iptables)

O firewall no `Router1` foi configurado com a política padrão da cadeia `FORWARD` como `DROP`. As seguintes regras de permissão foram adicionadas:

Listing 2: Script de regras do firewall no Router1.

```
# 1. Permite conexoes estabelecidas ou relacionadas
sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# 2. Permite HTTP da Internet (enp0s9) para a DMZ (enp0s8)
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -p tcp --dport 80 -j ACCEPT

# 3. Permite solicitacoes DHCP (de R2 na enp0s16 para DHCPServer na enp0s10)
sudo iptables -A FORWARD -i enp0s16 -o enp0s10 -p udp --dport 67 -j ACCEPT

# 4. Permite respostas DHCP (de DHCPServer na enp0s10 para R2 na enp0s16)
sudo iptables -A FORWARD -i enp0s10 -o enp0s16 -p udp --sport 67 -j ACCEPT

# 5. Define a politica padrao para BLOQUEAR
sudo iptables -P FORWARD DROP
```

As regras foram salvas permanentemente com o pacote `iptables-persistent`.

4 Resultados e Análise dos Testes

Foram realizados testes para validar a implementação.

4.1 Teste de Acesso HTTP (Permitido)

Foi executado o comando `curl http://10.0.20.1` a partir do `WebTermPublic`.

- **Resultado:** SUCESSO. A página de teste foi retornada.
- **Análise:** A regra do firewall para a porta 80 funcionou como esperado, permitindo o tráfego legítimo para o servidor web.

4.2 Teste de Bloqueio de Acesso à Rede Interna

Foi executado o comando `ping 10.0.30.1` a partir do `WebTermPublic`.

- **Resultado:** FALHA. O comando não recebeu nenhuma resposta.
- **Análise:** A política padrão `DROP` do firewall bloqueou com sucesso o tráfego ICMP (ping) não solicitado, protegendo a rede interna.

4.3 Teste de Funcionamento do DHCP

Foi executado o comando `sudo dhclient -v enp0s3` na `WebTermWorkstation`.

- **Resultado:** SUCESSO. A estação de trabalho recebeu um endereço IP ('10.0.40.103') do `DHCPServer`.
- **Análise:** O teste validou toda a cadeia de comunicação: o cliente, o serviço de relay no `Router2`, as regras de permissão de DHCP no firewall do `Router1` e o serviço de servidor no `DHCPServer`.

5 Conclusão

O projeto atingiu com sucesso todos os seus objetivos. Foi possível construir uma rede segmentada e segura, onde o tráfego foi meticulosamente controlado por um firewall stateful. A configuração demonstrou a proteção eficaz da rede interna contra acessos externos não autorizados, ao mesmo tempo em que permitiu a exposição controlada de serviços na DMZ e o funcionamento de serviços internos essenciais como o DHCP. Os desafios técnicos encontrados durante a implementação serviram para aprofundar o conhecimento sobre configuração de redes e diagnóstico de problemas em ambientes Linux.

6 Referências

1. Documentação Iptables. Disponível em: `https://www.iptables.org/documentation/HOWTO/pt/packet-filtering-HOWTO.sgml`
2. Exemplos de Regras Iptables. Disponível em: `https://linuxconfig.org/collection-of-basic-linux-firewall-iptables-rules`
3. Pacific Cyberlab - Lab 8: Firewalls. Disponível em: `https://cyberlab.pacific.edu/courses/comp177/labs/lab-8-firewalls`