Segurança Perimetral

Firewalls

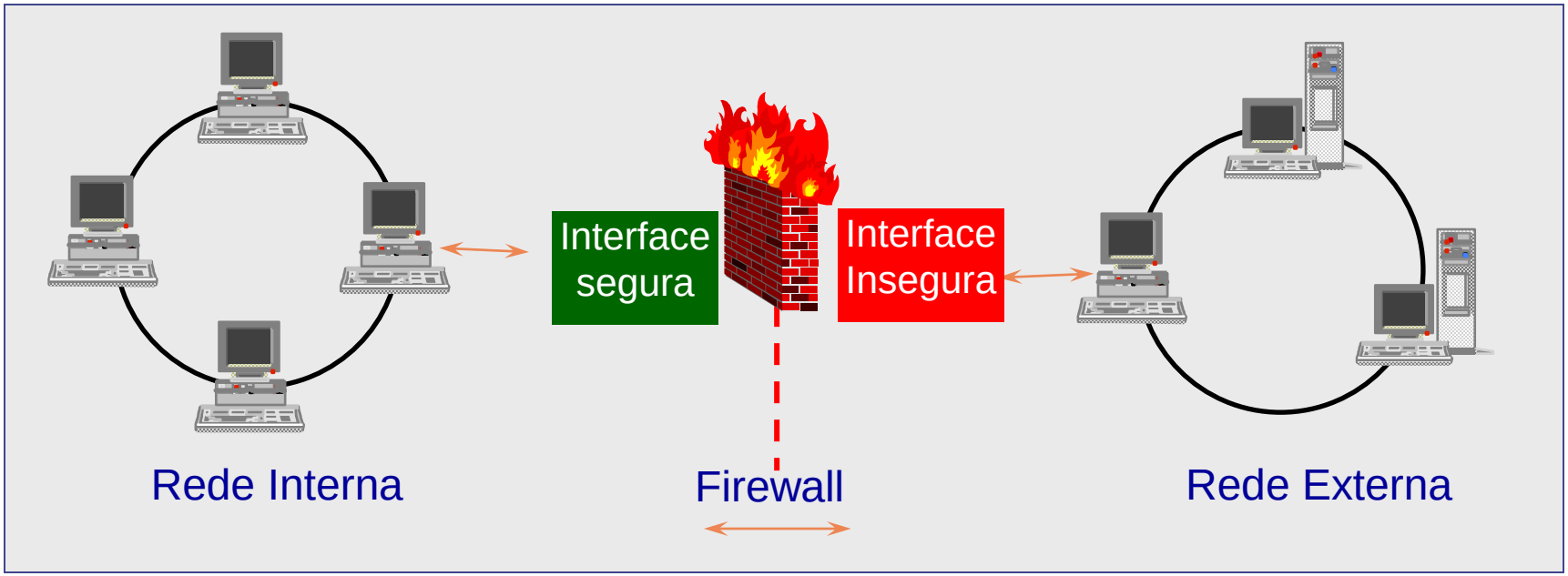
Dênio Mariz
Daniel Melo
denio@cefetpb.edu.br
daniel@codata.pb.gov.br

Nov/2009

Firewalls

- Motivação:
  - Proteger uma rede privada contra “intrusos”
  - Impedir acessos a recursos computacionais por usuários não autorizados
  - Impedir exportação de informações não autorizadas
- Outros Propósitos:
  - Bloquear acesso a sites particulares
  - Prevenir que certos usuários/máquinas acessem certos servidores/serviços
- Firewall é um mecanismo para controlar quais dados saem e entram na sua rede
- Quando bem configurados são difíceis de quebrar

O Sistema Firewall



- Consiste em uma máquina interceptando todo o Tráfego de Entrada e Saída da Rede
- Pode ser configurada para Filtrar acesso da Internet para a Rede e da Rede para a Internet

Tipos de Firewall

- 2 tipos de dominam:
  - Packet Filtering gateways
  - Proxies de Aplicação

Packet Filtering Gateway

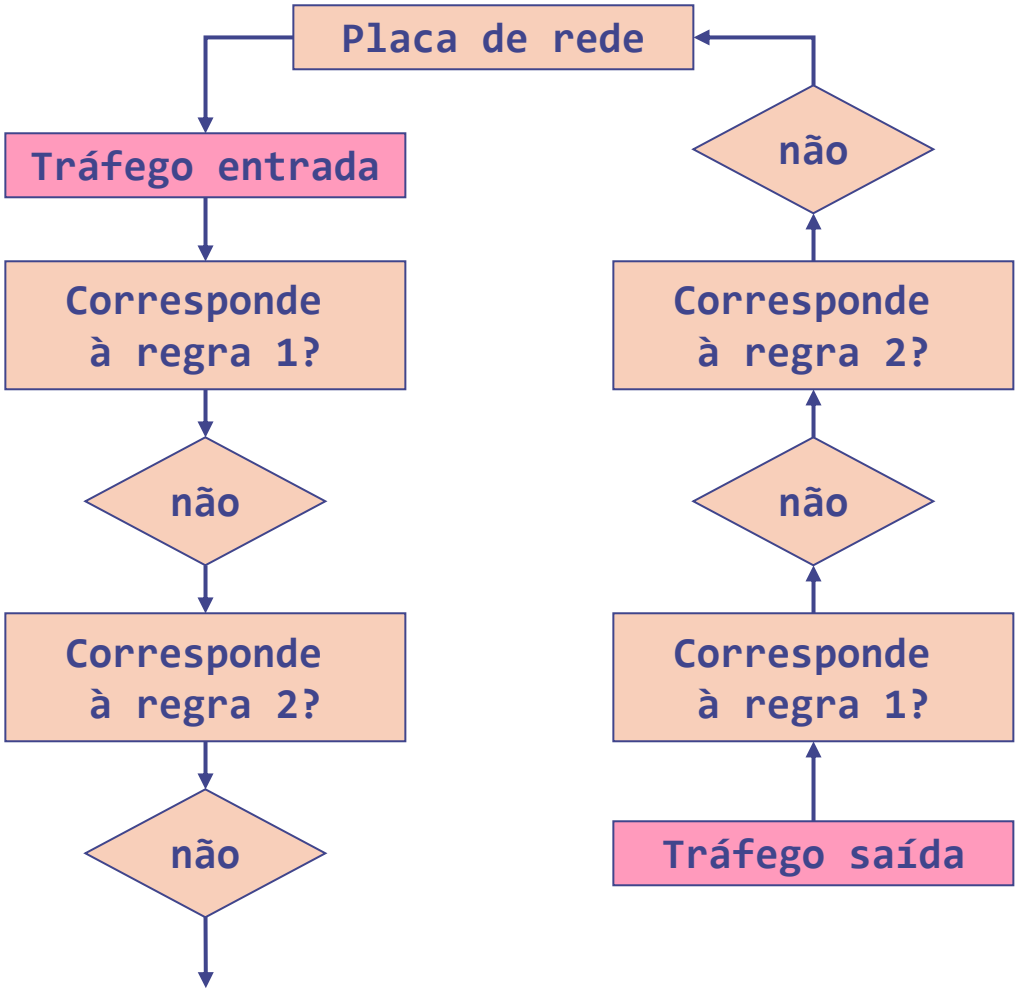
- Opera nos níveis de rede e transporte
- Packet Filter é configurado para filtrar pacotes que passam na interface do roteador
  - Endereços IP (geralmente destino)
  - Porta TCP/UDP para certos hosts internos
  - Pode se basear em regras mais complexas
- Vantagens
  - Baixo custo
  - Flexibilidade
  - Simplicidade
  - Boa performance

Application Proxies

- Atuam no nível de aplicação
- Alguns exemplos:
  - Podem bloquear ou redirecionar URLs em função de uma “lista negra”
  - Registrar URLs acessadas
  - Podem filtrar, bloquear ou responder e-mails
  - “E-mails com arquivo atachado somente se compactado”
- Fluxo de saída pode ser forçado a passar por um proxy
- Limitação de performance
  - Vasculhar dentro de pacotes pode ser demorado quanto o tráfego é alto

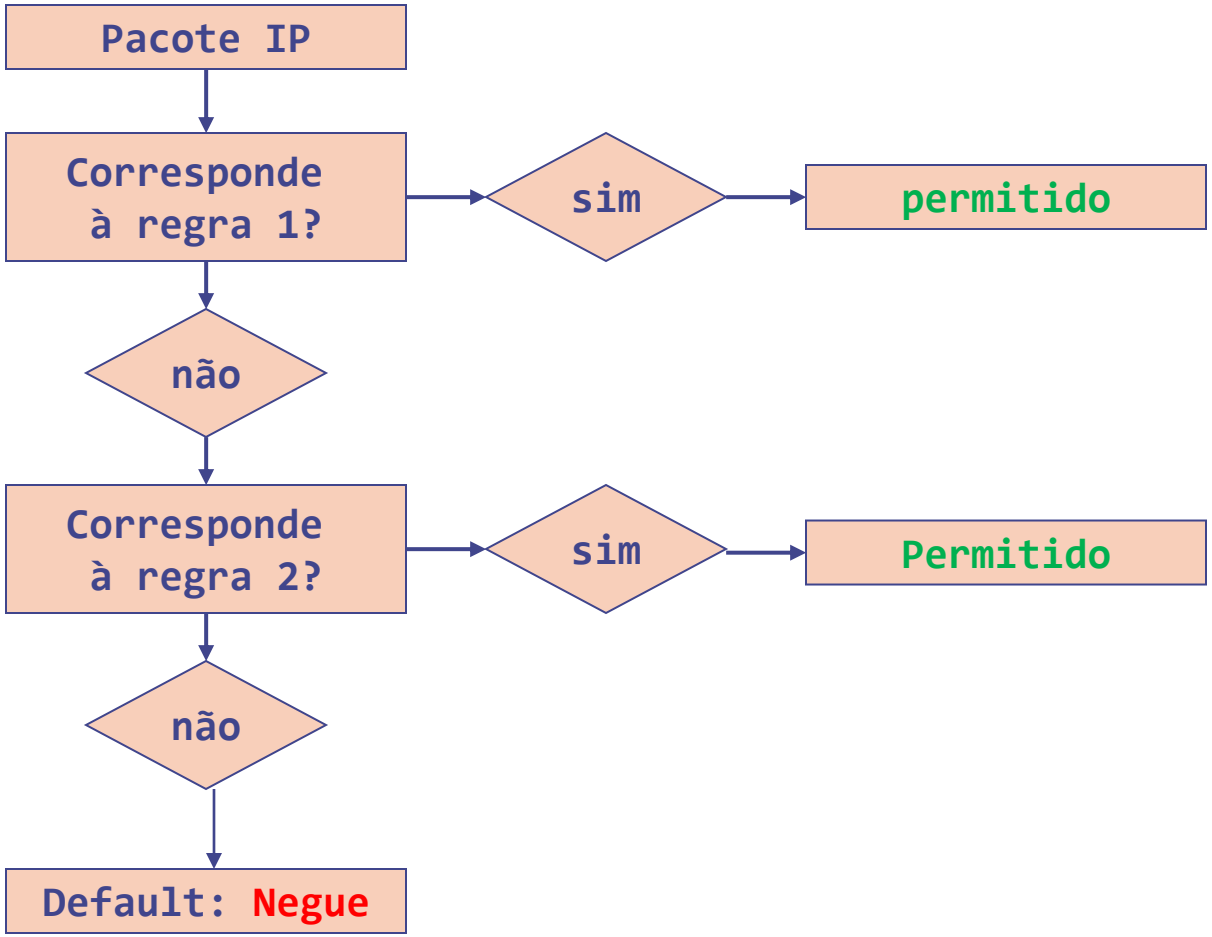
Regras de Filtragem

➔ Os tráfegos de entrada e saída devem ser filtrados separadamente.



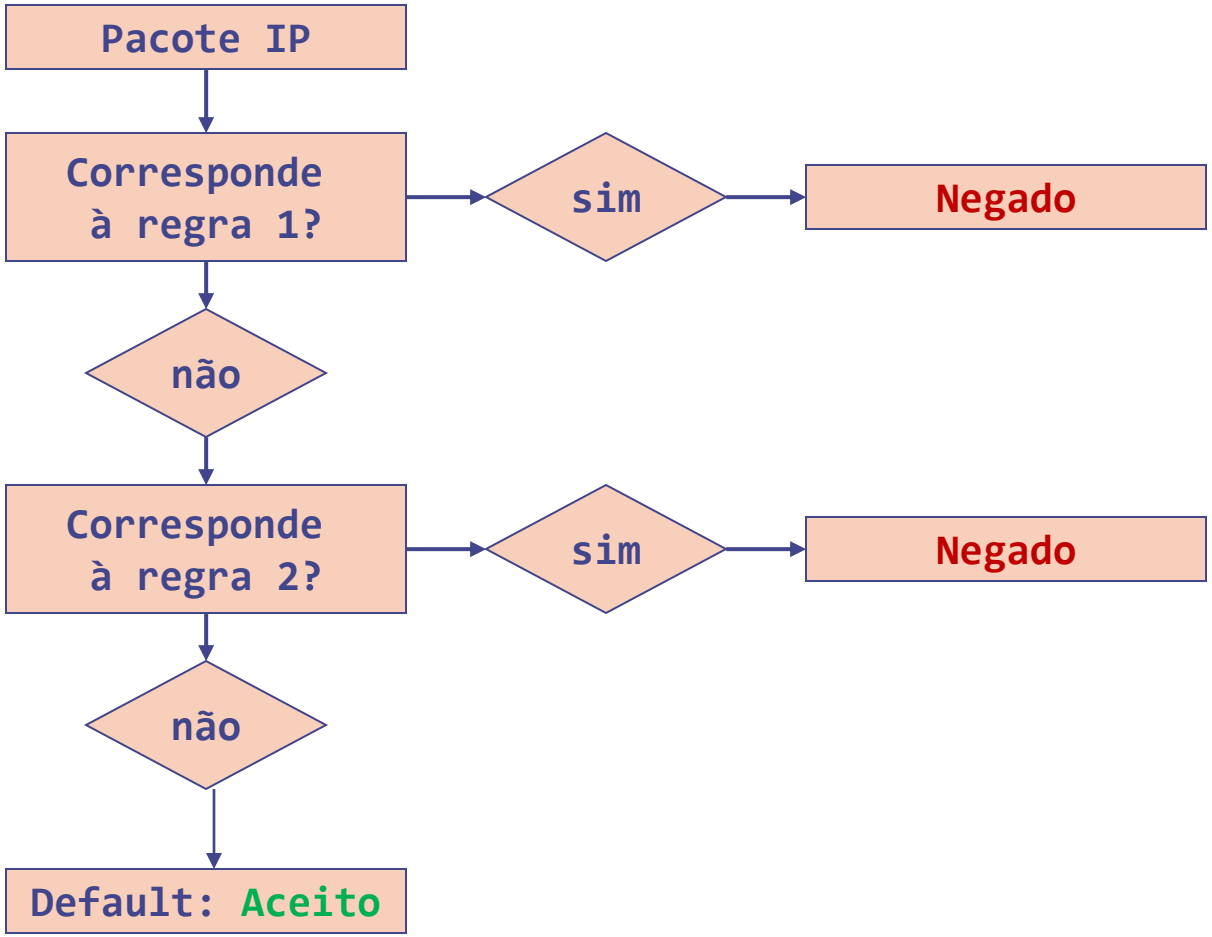
Regras de filtragem (2)

➔ Política default: negue tudo (**default deny**)



Regras de filtragem (3)

➔ Política default: Permita tudo (**default permit**)



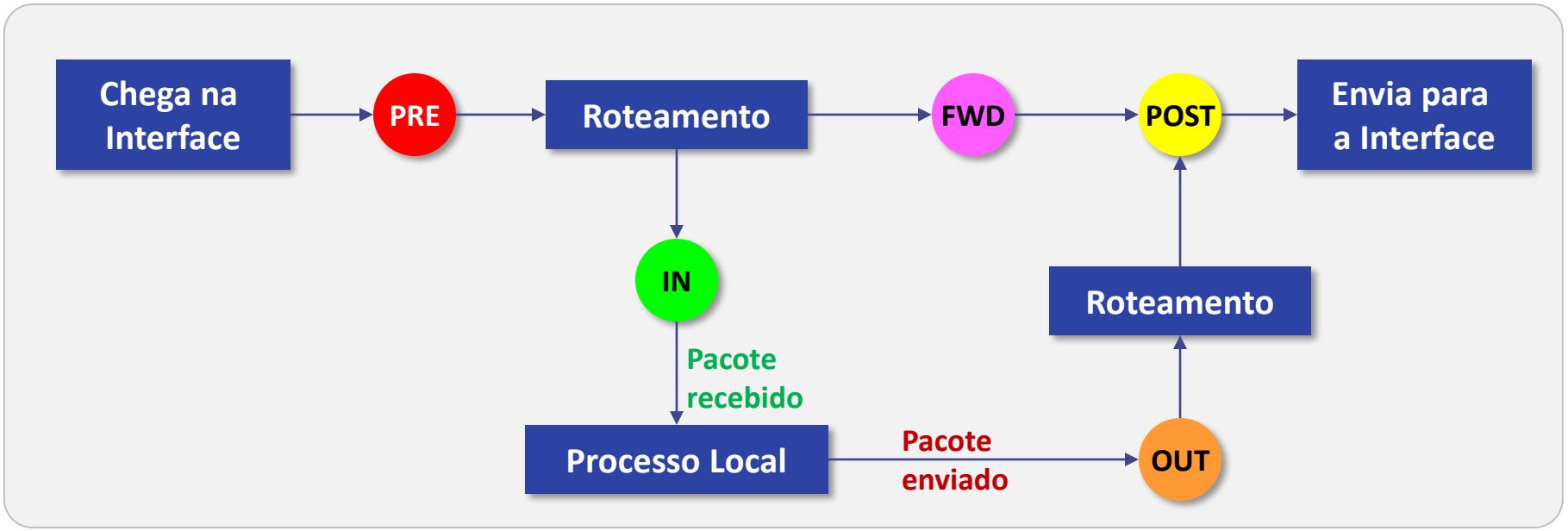
Linux Firewalls

IPtables

IPtables

- Sistema de seleção de pacotes construído sobre o framework Netfilter.
  - Define pontos na passagem de pacotes pela pilha de rede
  - Pontos de passagem são acionados pelo Kernel sempre que um pacote passa por esses pontos
- Constituído por tabelas de regras, por onde os pacotes deverão passar, durante a passagem nos pontos da pilha de rede:
  - Filter – tabela default utilizada para filtragem de pacotes
  - NAT – tabela usada para Network Address Translation
  - Mangle (mutilação) – alterações especializadas nos pacotes

Pontos de “captura” do IPtables



- PRE ---> Mangle e NAT
- IN ----> Filter e Mangle
- FWD ---> Filter e Mangle
- OUT ---> Filter, NAT e Mangle,
- POST --> Mangle e NAT

Iptables (3)

- Cada tabela é constituída por **chains** (encadeamentos)
  - Pré-definidas no Iptables ou criadas pelo usuário.
- Tabela Filter
  - INPUT
  - OUTPUT
  - FORWARD
- Tabela NAT
  - PREROUTING
  - OUTPUT
  - POSTROUTING
- Tabela Mangle
  - PREROUTING
  - OUTPUT

A tabela “filter”

- `iptables -t filter Comando [regra] [opções] [-j Alvo]`
- `-t filter`
  - Indica a tabela filter
- `Comando`
  - Operações com as chains (INPUT, OUTPUT, FORWARD)
  - Inserir, remover, adicionar, ..., uma regra
- `Regra`
  - Indica o que deve ser observado no pacote
  - Pode ser visto como uma expressão lógica
- `Opções`
  - Detalhes para estabelecimento da regra
  - Campos do cabeçalho etc
- `Alvo`
  - O que deve ser feito com o pacote

Comandos 1/2

- -F chain
  - Apaga conteúdo da chain, ou de todas as chains se nenhuma for especificada
  - iptables –t filter –F INPUT
  - iptables –t filter –F OUTPUT
  - iptables –t filter –F FORWARD
- -L chain
  - Lista as regras de uma chain, ou de todas as chains se nenhuma for especificada
  - -L –v
    - **Lista informação adicional (verbose)**
  - `-L –n --line-numbers`
    - **Lista a posição das regras na chain.**
- Exemplo
  - iptables –t filter –L –n –line-numbers

Comandos 2/2

- -A <chain>
  - Adiciona uma regra no fim da chain
- -I <chain> <num regra>
  - Insere uma regra antes da regra <num_regra>
- -D <chain> <num regra>
  - Apaga a regra na posição <num regra>
- Exemplos
  - iptables –t filter -A INPUT ....
  - iptables –t filter -I INPUT 2 ....
  - iptables –t filter -D OUTPUT 2

Regras

- -p <protocolo>
  - Protocolo pode ser: tcp, udp, icmp, all
  - Opções estendidas
    - **-p tcp --sport 80 (porta fonte)**
    - **-p tcp --dport 22 (porta destino)**
    - (outras)
- -s IP
  - Endereço de origem
- -d IP
  - Endereço de destino
- -i input_interface
  - Interface de entrada

Exemplos

➔ # Libera tráfego vindo da Interface confiável:
➔ **iptables –t filter -A INPUT -i eth1 -j ACCEPT**

➔ # Libera de Acesso SSH ao firewall:
➔ **iptables –t filter -A INPUT -p tcp --dport 22 -j ACCEPT**

➔ # Liberacao de acesso WEB ao firewall:
➔ **iptables –t filter -A INPUT -p tcp --dport 80 -j ACCEPT**

➔ # Libera a interface de loopback
➔ **iptables –t filter -A INPUT -s 127.0.0.1 -j ACCEPT**

➔ # Substitui a regra 2 por outra
➔ **iptables -R INPUT 2 -d 127.0.0.1 -p icmp -j DROP**

Exemplos

➔ # Libera entrada de pacotes ICMP
➔ **iptables –t filter -A INPUT -p icmp -j ACCEPT**

➔ # Libera roteamento quando chega pela interface confiável
➔ **iptables –t filter -A FORWARD -i eth1 -j ACCEPT**

➔ # Libera roteamento de trafego ICMP
➔ **iptables –t filter -A FORWARD -p icmp -j ACCEPT**

➔ # nega saida de tráfego TELNET
➔ **iptables –t filter -A OUTPUT -p tcp --dport 23 -j REJECT**

➔ # Não roteia da rede 192.168.1.0/24 para a rede 192.168.2.0/24
➔ **iptables -A FORWARD –s 192.168.1.0/24 –d 192.168.2.0/24 -j REJECT**

DROP vs REJECT

- DROP descarta o pacote
- REJECT envia um pacote de volta ao remetente
  - TCP: RST/ACK
  - UDP: ICMP destination port unreachable
- Opções para REJECT
  - TCP: -p tcp -j REJECT --reject-with tcp-reset
  - UDP: -p udp -j REJECT --reject-with icmp-port-unreachable

Exercícios