

# Network Security: Firewalls & VPNs

SIO

André Zúquete

**deti** universidade de aveiro  
departamento de eletrónica,  
telecomunicações e informática

# Firewalls: Objectives

1995 ~

// → Antigamente era para  
proibir o acesso à  
internet dentro da empresa

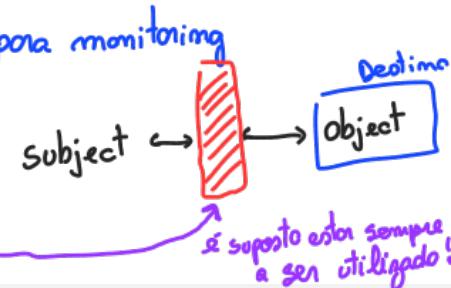


- Fundamental element to interconnect network domains
  - Access control
  - Flow control
  - Content control → Malwares, ...
- Centralized implementation of security policies
  - Minimizes the impact of local vulnerabilities
    - Known or unknown
  - Makes it easier to take more drastic actions
  - Centralizes problem detection
    - and its treatment→ Mais canga!

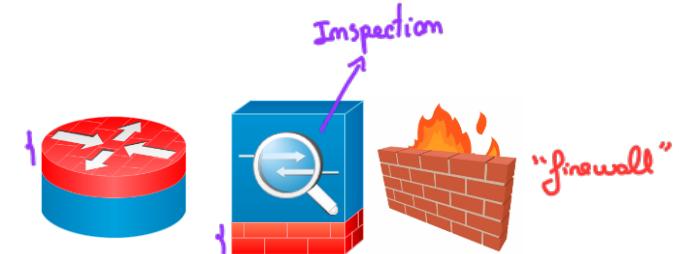
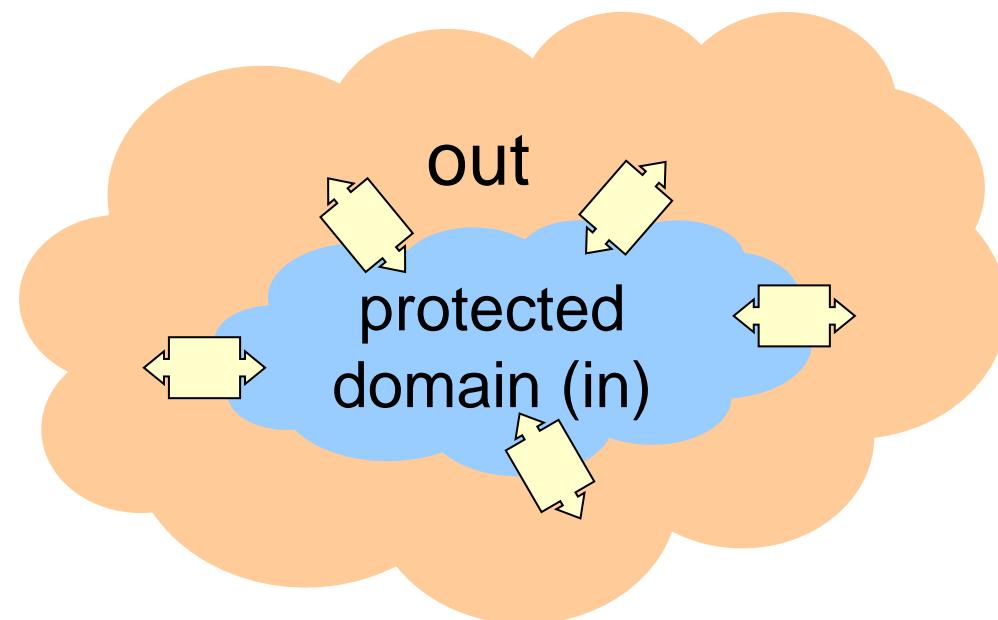


# Firewall definition (Cheswick & Bellovin)

- Link between network domains
  - of a protected perimeter (set of networks and machines)
  - to an insecure network ← máquinas infetadas,
    - Internet
    - Other untrusted local networks → Posso não confiar em redes da própria empresa!,,
- Component set
  - Hardware and software
- Properties
  - In the path of all in ⇔ out traffic
    - Controls the traffic passing through it
  - Immune to penetration (by definition) ← se suposto estiver sempre a ser utilizado!



# Firewalls: Definition (Cheswick & Bellovin)



# Firewalls: Functionalities

- **Supervision of all in ⇔ out communication**

- Control
  - The use of internal resources by external hosts/requests
  - The use of external resources by internal host/requests
- Defense from attacks
  - from outside the protected domain towards its resources
  - from the protected domain against external resources

DNS:  
1.1.1.1  
8.8.8.8  
↳ é barrado pela VA  
• Tô também não é permitido!,,

- **Activation of gateway mechanisms**

- To hide the structure from the protected perimeter

- NAT (Network Address Translation)  
↳ Super importante para a segurança!,,  
↳ Tradução: (privada : pública)
- Masquerading and Port Forwarding

- To extend the security perimeter

- Secure tunneling (VPN)

Servem para estender o perímetro de segurança a um cliente!

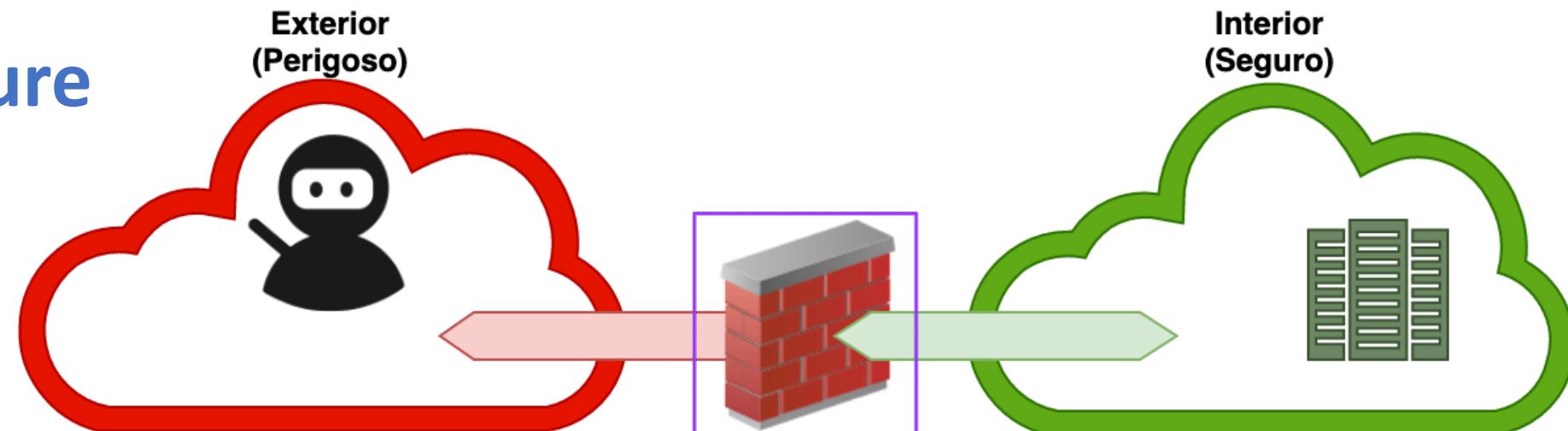
Posso ter um porto exposto numa máquina com end. público ...

# Firewalls: Importance

- Extreme!
- Attacks on public systems are constant,
  - By specialized attackers
  - By standalone applications
- Systems do not always have adequate security mechanisms
  - Blocking after too many incorrect attempts
  - Validation of communications
    - Access control
- Necessary to apply mechanisms defined by the administrator, in accordance with domain policies
  - An application programmer is not aware of these

# Generic structure

Defesa de perímetro



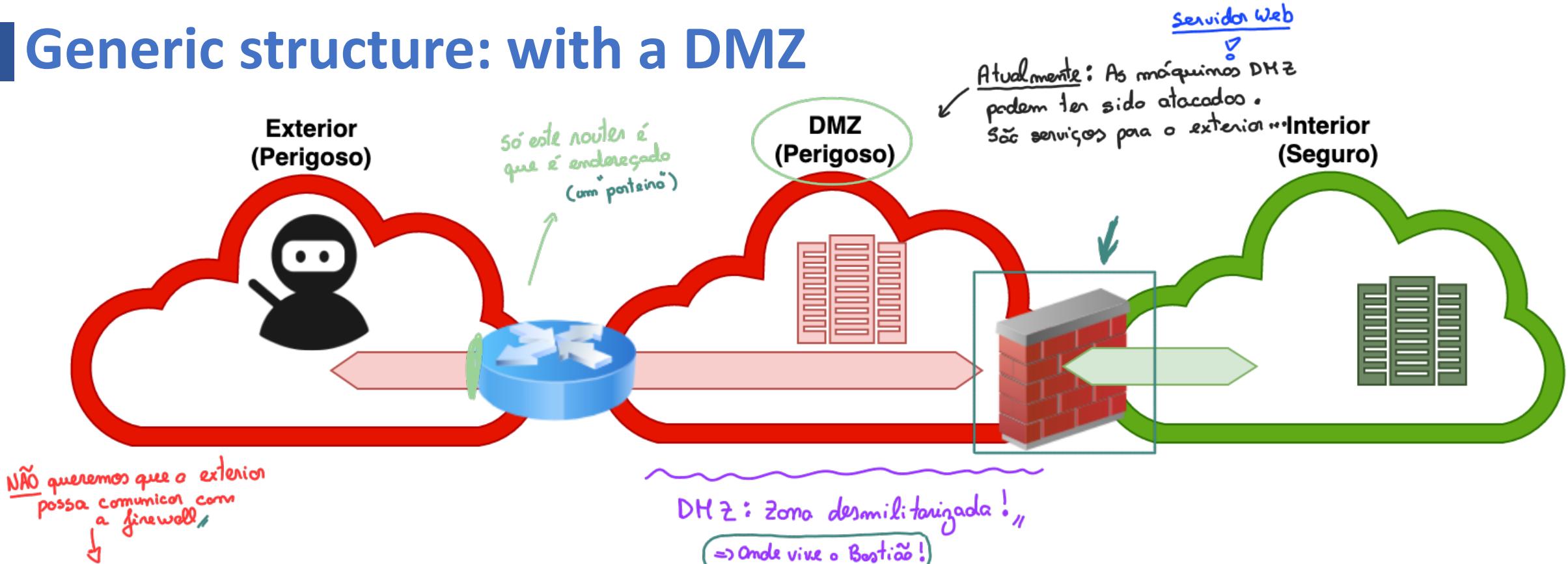
Type 1,,

- Perimeter defense (of the domain)
  - Can be part of a defense in depth strategy
- Consider an unsafe environment and a safe one
  - Out: other domains or the Internet
  - Inside: internal network
- A single server: Bastion

• Controle o tráfego  
• Bem atualizado  
• (...)

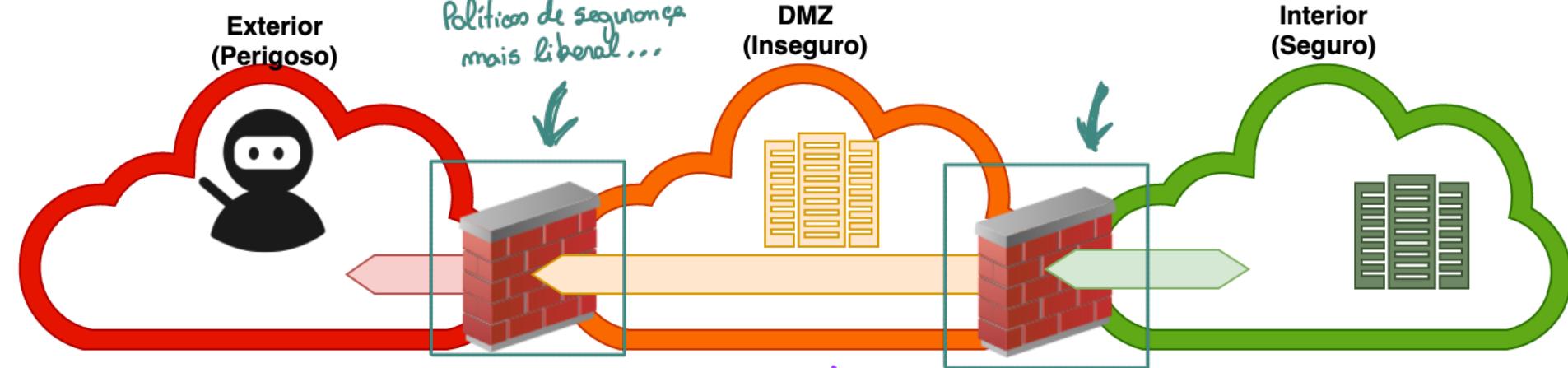
↑ Não devemos mais poder ligar  
ao Bastião da internet

# Generic structure: with a DMZ



- **DMZ: DeMilitarized Zone Network or Perimeter Network**
  - Insecure network
  - Contains servers exposed to the world
  - Sometimes necessary to use specific services/applications

# Generic structure: w/ DMZ and two firewall equipments



- **DMZ may have some protection**

- System of two firewalls with different rules

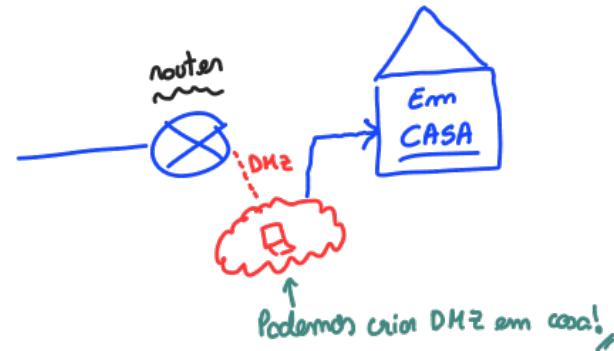
(+) Corre!

- External firewall: quite permissive

- Control access to all networks

- Internal firewall: more restricted

- Control access to the internal network



# Firewall types: Packet filters



Packet headers

- Reject unauthorized interactions based on the content of IP datagrams

- IP addresses (source and/or destination)
- IP/transport header options
- Transport protocols and ports (origin and/or destination)
- Directions for creating virtual circuits
- Data sent via transport protocol
- Datagram size

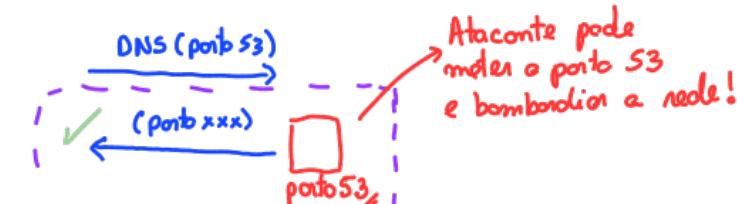
- Can analyze flow behavior
  - Example: detect port scans (with nmap)

- Typically supported by core OS components
  - Example: iptables, ipfw, pf

• Port 443: usou ssh  
https

BLOQUEAR

Problema: As firewalls têm pouca memória ...



Port 53 como origem ele aceita ...  
→ Termos de guardião estado!

# Firewalls types: **Stateful packet filters**

*Regras que usam estado!,,*

- Dynamic (or context-sensitive) packet filter
  - Sort of packet filter with historical context
  - Context is key to certain decisions
  - Common term: Stateful Packet Filter/Inspection (SPI)
- Context examples:
  - Decisions made for IP packet fragments
    - Defragmentation before filtering  
*Desfragmentar antes!*
  - Established TCP virtual circuits
    - Circuit establishment requests are controlled
    - Established virtual circuits are allowed



# Firewall types: Stateful packet filters

- Context examples (cont.):

- Dynamic NAT tables

- Creation of entries depending on observed traffic

- Request/response interactions over UDP

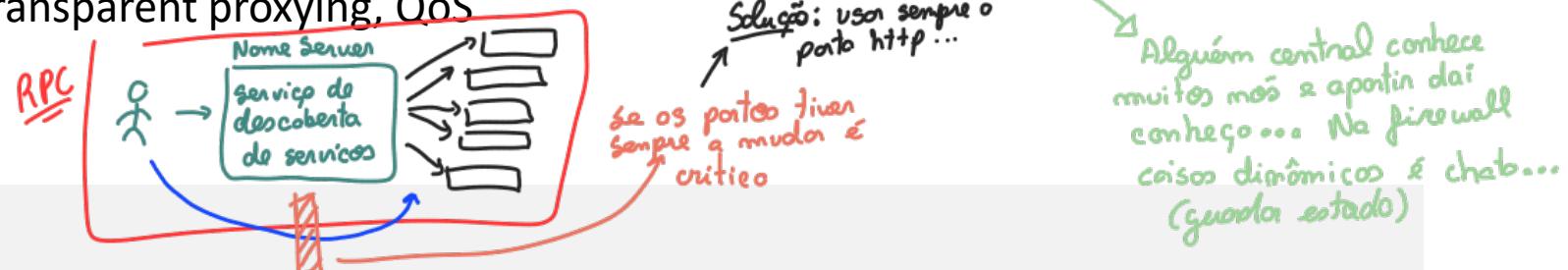
- Dynamic authorization of responses to authorized requests
  - Example: DNS name resolution

- ICMP error messages

- Related to previously sent TCP/UDP packets

- Identification of application protocols from data flows

- To handle flows that use dynamic or “stolen” ports
  - Examples: FTP, RPC protocols, P2P protocols  
*→ chama a uma função dentro de um serviço!*
  - Utility: filtering, transparent proxying, QoS

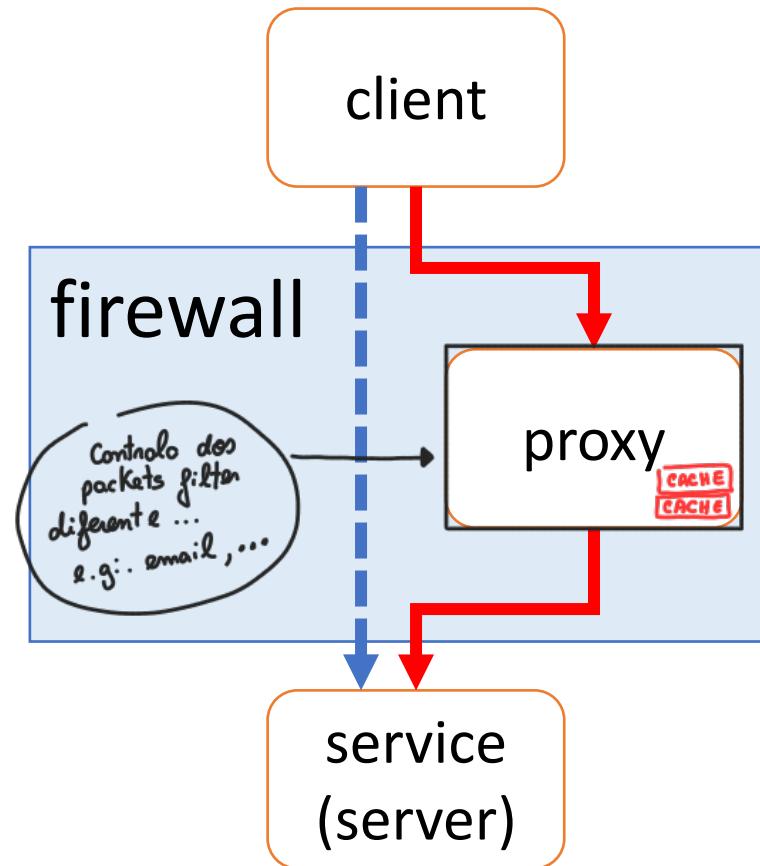


# Firewall types: Applicational gateways!

↳ Fluxo aplicacional!

- Control interactions at the application level
  - But transparent to interacting applications !
  - There is usually a different firewall per protocol (protocol proxy)
- Client -> Proxy -> service (server)
  - Proxies are servers
- Aspects of operating a proxy
  - User access control
  - Analysis and modification of content
  - Detailed logging !, → tenho uma missão aplicacional !,
  - Impersonation (proxying)
    - Transparent replacement of one of the interlocutors

NOTA: esta firewall (proxy)-Application gateway é colocada do lado do servidor !!



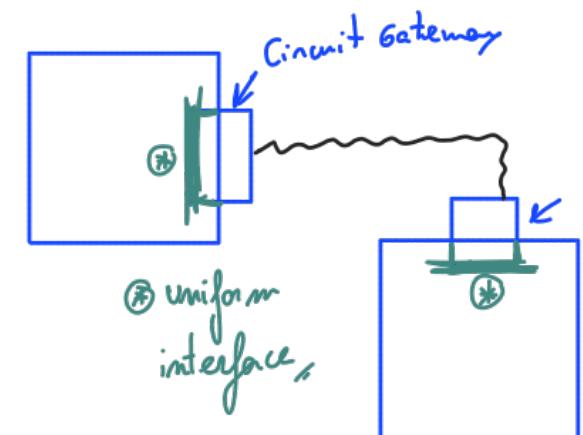
# Firewall types: Circuit gateways

- Kind of application gateway
  - But contacted directly by client applications
  - Non-transparent interposition
    - ↳ O cliente liga-se à Firewall !!!,,
    - ↳ O cliente usa voluntariamente à proxy...
- Interposition goals
  - Deploying domain-specific authentication and authorization policies and mechanisms
  - Deploying supplementary services
    - e.g. Tor proxy ↳ é voluntariamente redirecionado (o próprio browser) para o endereço do Tor proxy...
- Typically requires changing client applications
  - Examples: SOCKS and HTTP Proxy

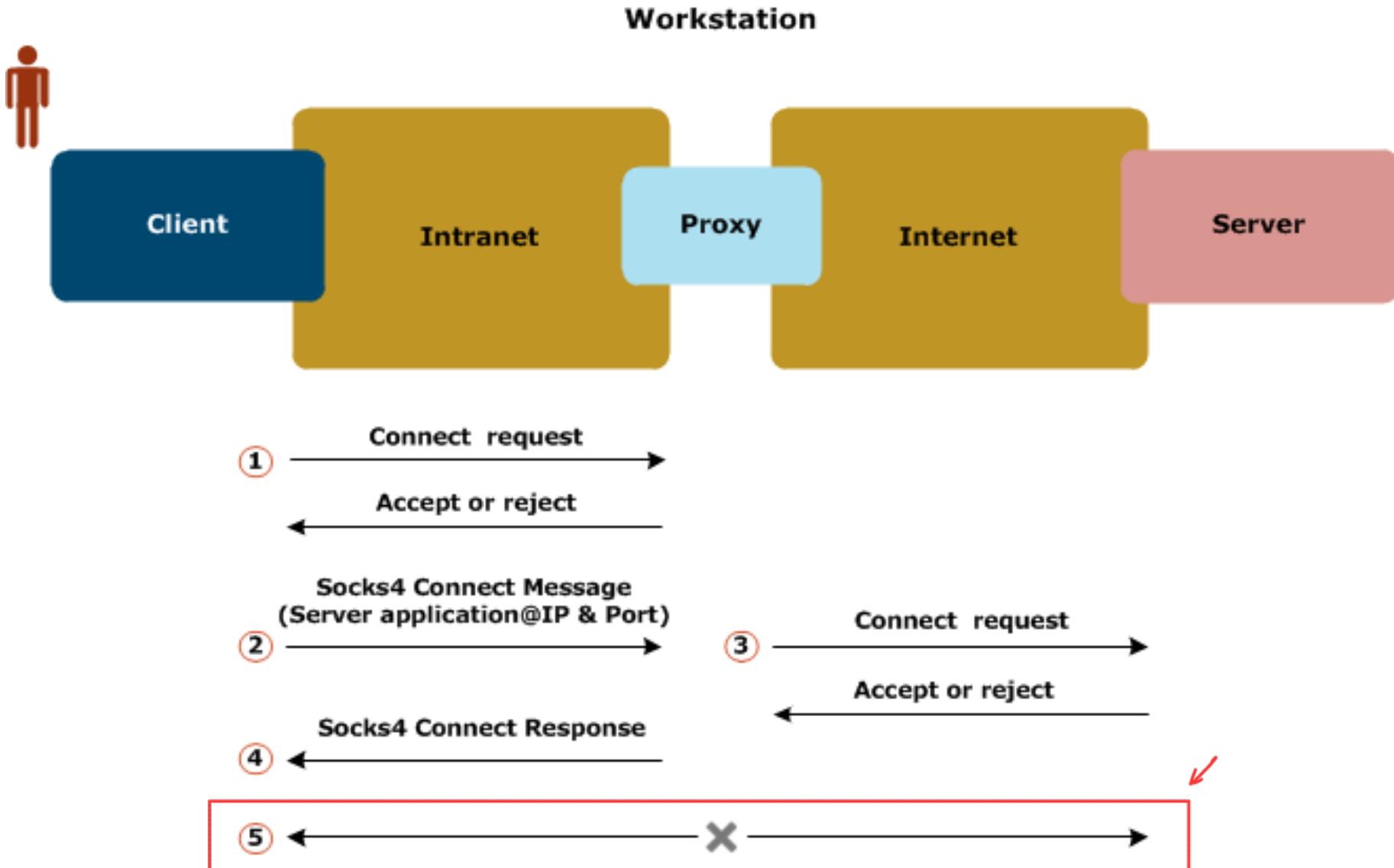
Circuit gateway é colocada do lado do Cliente !!

O Cliente, voluntariamente, liga-se ao proxy. Escolhe, ao nível aplicacional, o servidor que se quer ligar. O proxy liga-se e vai reencaminhando (com ou não alterações) e ajuda a comunicação Cliente ↔ Servidor

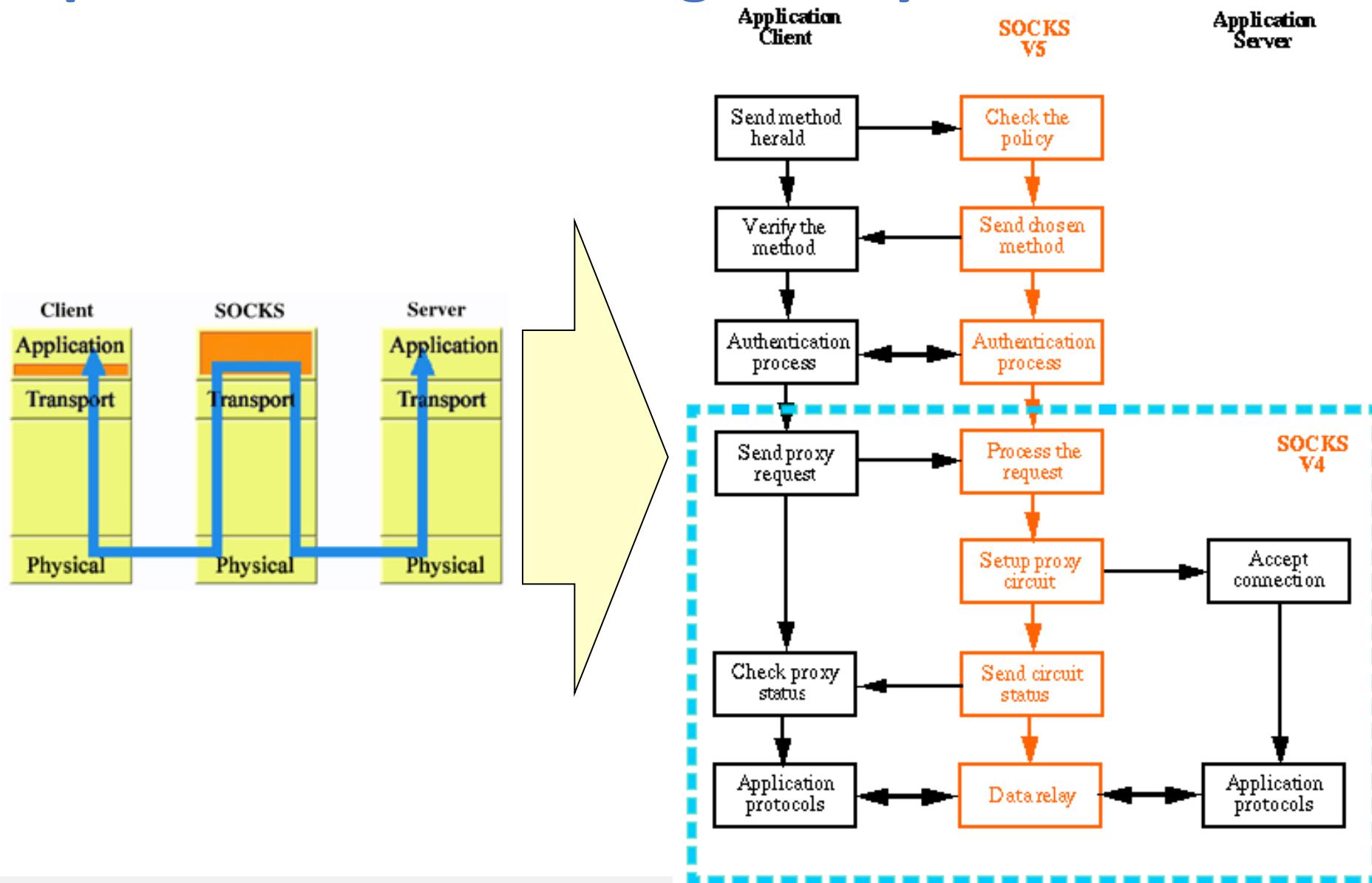
Protocolo SOCKS4



# Example: SOCKS4 circuit gateways

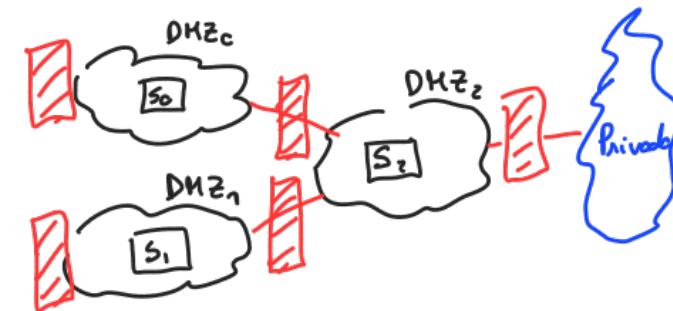


# Example: SOCKS v5 circuit gateways



# Firewall bastion

- Must run secure versions of operating systems
  - With a secure configuration
  - Only essential services are installed
  - Telnet/SSH, DNS, FTP, SMTP and authentication proxies
- Public servers should not perform in a bastion
  - Examples: DNS, SMTP, HTTP, FTP, SSH, RAS, etc. → Máquinas dedicados por cada rede
  - Must run on isolated machines within DMZs
    - Preferably one per service
  - Bastion only forwards traffic to the appropriate machines on a DMZ
    - And allows limited traffic from the DMZ



# Firewall bastion

- It is often a platform for application gateways
  - But the more proxies, the lower its performance will be
  - Proxies can run on specific machines
    - Security appliances → *Maquina com funcionalidade específica na área da segurança*
  - Bastion only forwards traffic to and from the appliances
- Secure execution of application gateways
  - Independence
    - The compromise of one does not affect the rest
  - No special privileges
    - Their compromise does not allow to affect the host

- Authorization

- Data streams (packet filters)

- Transport or network level  
*Não servem*

- Users (application gateways / circuits)

*Com base no user!*

- Traffic Redirection

- For dedicated hosts

- Local services (e.g., mail, www, ftp, etc.)
- Proxies in security appliances

- Proxying

- Explicit (e.g., circuit gateways)  
*e.g.: Tor proxy*

- Transparent (e.g., NAT address translations)

*↑ Abstração!*

# Firewalls' security services

- Application content processing
  - Content analysis
    - Example: virus detection
  - Changing high-level protocols
    - Example: virus removal
- Secure communication
  - Virtual Private Networks (VPNs)  
*...extender uma área...*
    - Encryption and integrity control of data flows over public (insecure)
  - Tunneling
    - IP domain extension to distant nodes
    - Example: PPTP, L2TP, IPSec

# Firewalls' security services

## • Defense against DoS attempts

- Attack detection
  - Abnormal traffic volumes, high volume, etc...
- Traffic scrubbing
  - Filtering dangerous or malformed datagrams
- Activation of mitigation add-ons
  - Example: SYN flooding relay/semi-gateway

ou DDoS  
Barrar todo os ip's  
 vindos do estrangeiro  
(...)

mmap!  
ele envia pacotes malformados e identifica ...

- Tentam fazer flooding!
- Impacto temporal!  
→ parar os serviços da UA durante o teste ...

## • Defense against information leaks (exfiltration attacks)

- Abnormal traffic detection
- Controlling behavior against known models



# Firewalls' limitations

- Cannot tackle attacks from the internal network
  - Unless the internal network is segmented into multiple subnets
  - Switches typically do not support firewall operations
  - VLANs provide minimal segregation (DMZ type)
- Effectiveness in controlling all external connections
  - Which can be done in parallel in countless ways:
    - Unregistered WLANs & Aps
    - VPNs
- Lack of control over camouflaged/hidden interactions
  - Camouflaged interactions multiplexed by VPNs
  - IP tunnels over HTTP, ICMP, DNS, etc.
- Difficult to manage in environments with heterogeneous interests
  - Universities, ISPs

TCP por cima de DNS  
→ camuflar os pacotes...



# Personal firewalls

- Adopted for the protection of individual / personal hosts
  - Defense in depth vs. perimeter defense
    - ↳ Defesa em cada máquina
    - ↳ ma ligação à rede
- Owners can set additional control policies
  - Applications authorized to access the network
  - The protocols that applications can use
  - The hosts/networks that protocols/applications can interact with
- Reduce the risk of compromise between hosts on a network
  - Allows a machine to self-protect, independently of the protection given by its network
    - No assumptions regarding other network protections

→ A rede do café pode ser // → Podemos ter  
imsegura... a mossa firewall
  - Useful for machines that migrate between networks

# Personal firewalls: issues

"Os mais idosos não sabem o que é TCP..."

- Normal users are not network security experts
  - They don't normally understand how IP networks work
    - IP addresses, transport ports, transport protocols, etc.
  - They do not know how to assess whether a given interaction is normal, acceptable, etc.
  - They don't know the basic security policies they should apply
- Blocking suspicious interactions may nullify functionality
  - Network communication is currently commonplace
  - Applications do not inform users of their communication needs

# Personal firewalls: issues

- Operational complexity

- Different operating environments → different policies
- Different network interfaces → different policies

→ Políticos para redes diferentes!

Domestic: X, Y  
Work: Z  
Public: H, I

→ Nos secalhar precisamos de mais granularidade



- The combination of operational scenarios, network interfaces and acceptable interactions for each case leads to a huge number of rules
  - Confusion, incoherence → difficulty to detect vulnerabilities

# Linux OS firewall: iptables

Firewall do Linux //

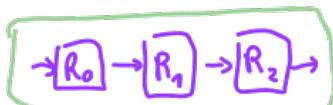
- Stateful packet filter

- Integrated with the kernel TCP/IP
- Can be extended in several ways
  - New core modules
  - User mode applications

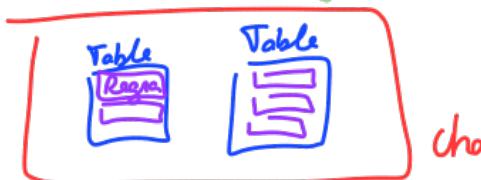
Firewalls Packet Filter: → Percorre todos os regras e verifica o padrão. Se verificar tomo decisão

- Regra:

- PADRÃO → encontram pacotes compatíveis
- DECISÃO → Caso haja padrão, executa a decisão



Chain é um conjunto de regras/Tables



- 5 chains

- INPUT, OUTPUT, FORWARD
- PREROUTING, POSTROUTING

- 4 tables (per chain, but not for all)

- raw, mangle, nat, filter

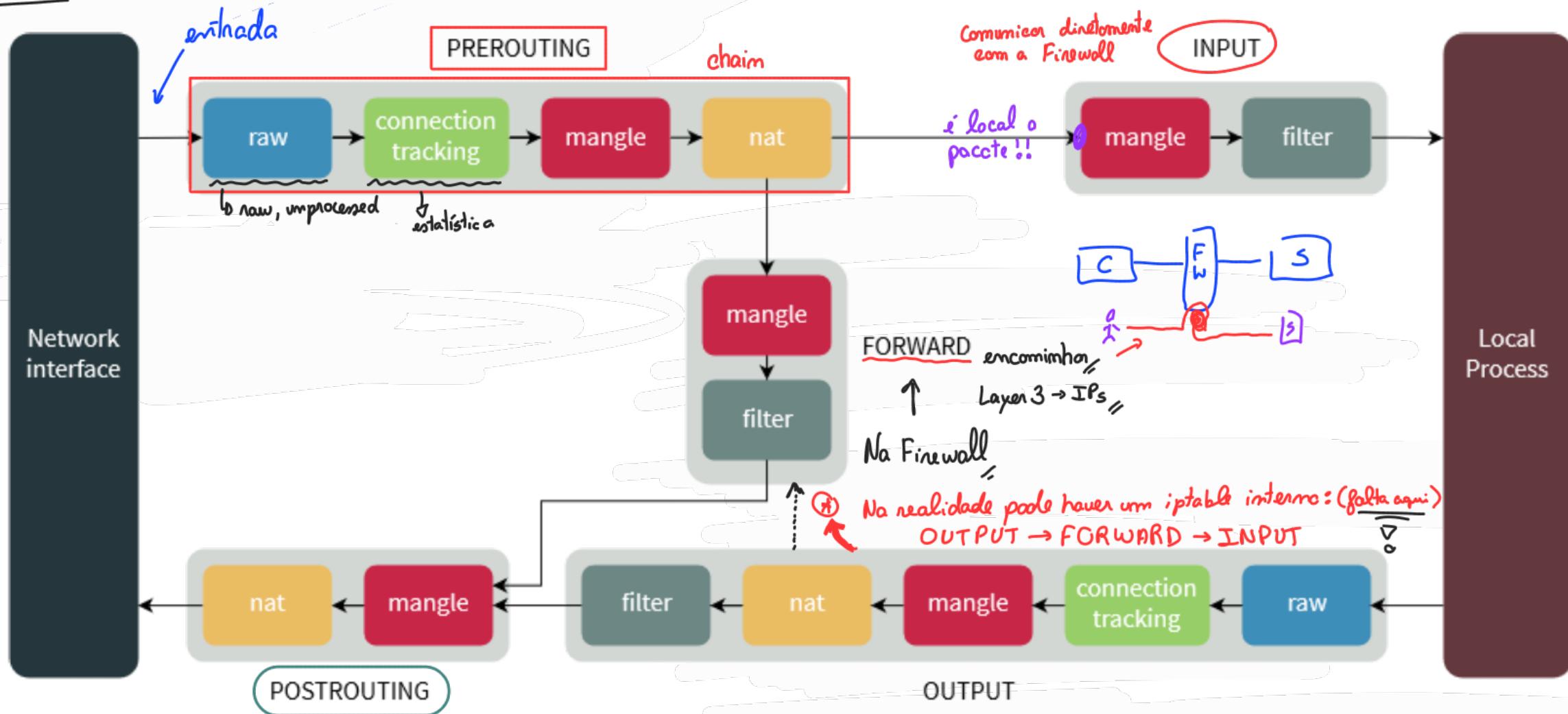
Decisões finais ou intermediárias

ou regras  
determina modificações no pacote  
"tiran"

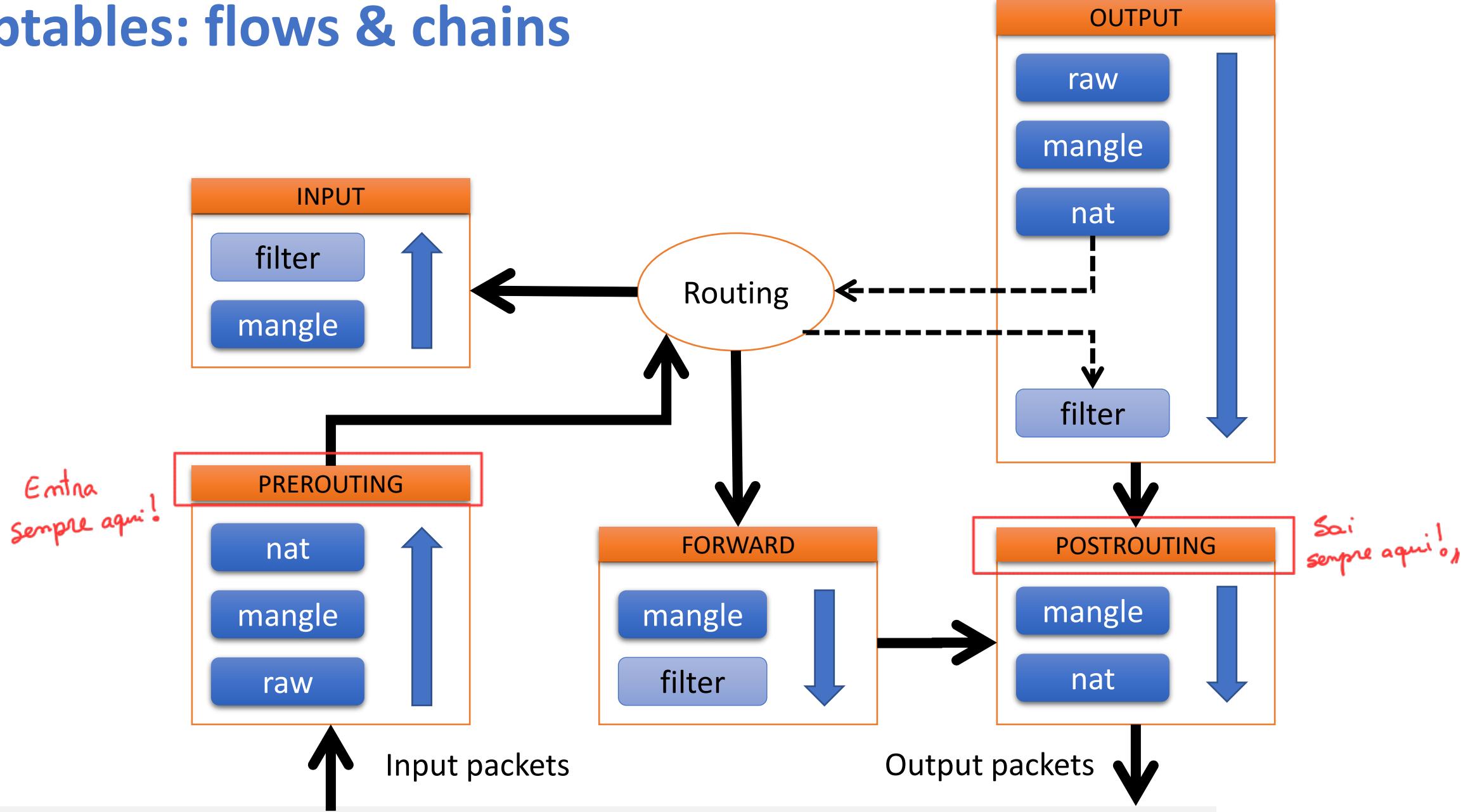
- Various extra modules

- e.g., CONNTRACK (connection tracker, or flow follower)

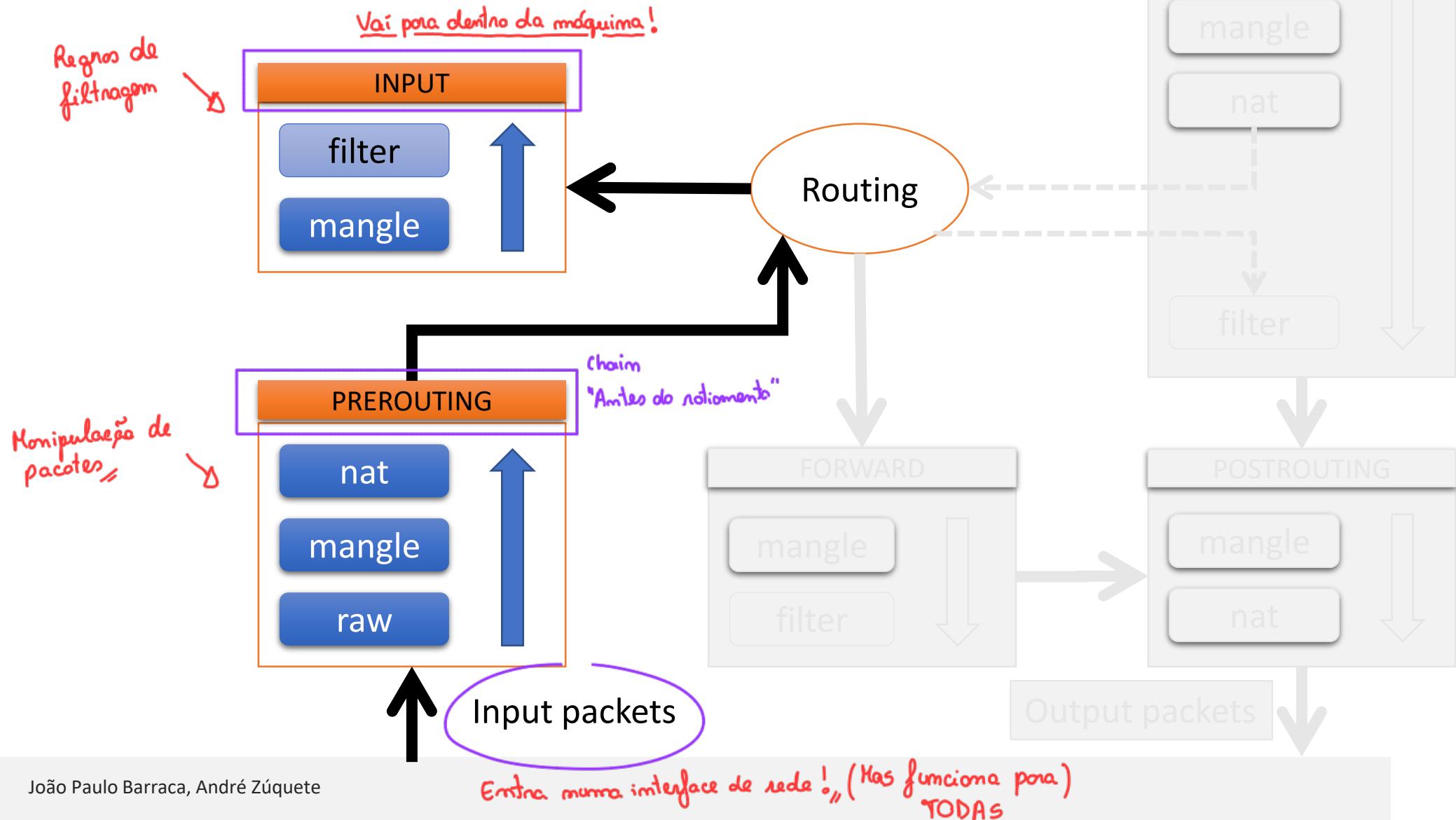
EXTRA: (aula prática) KELHOR GRÁFICO



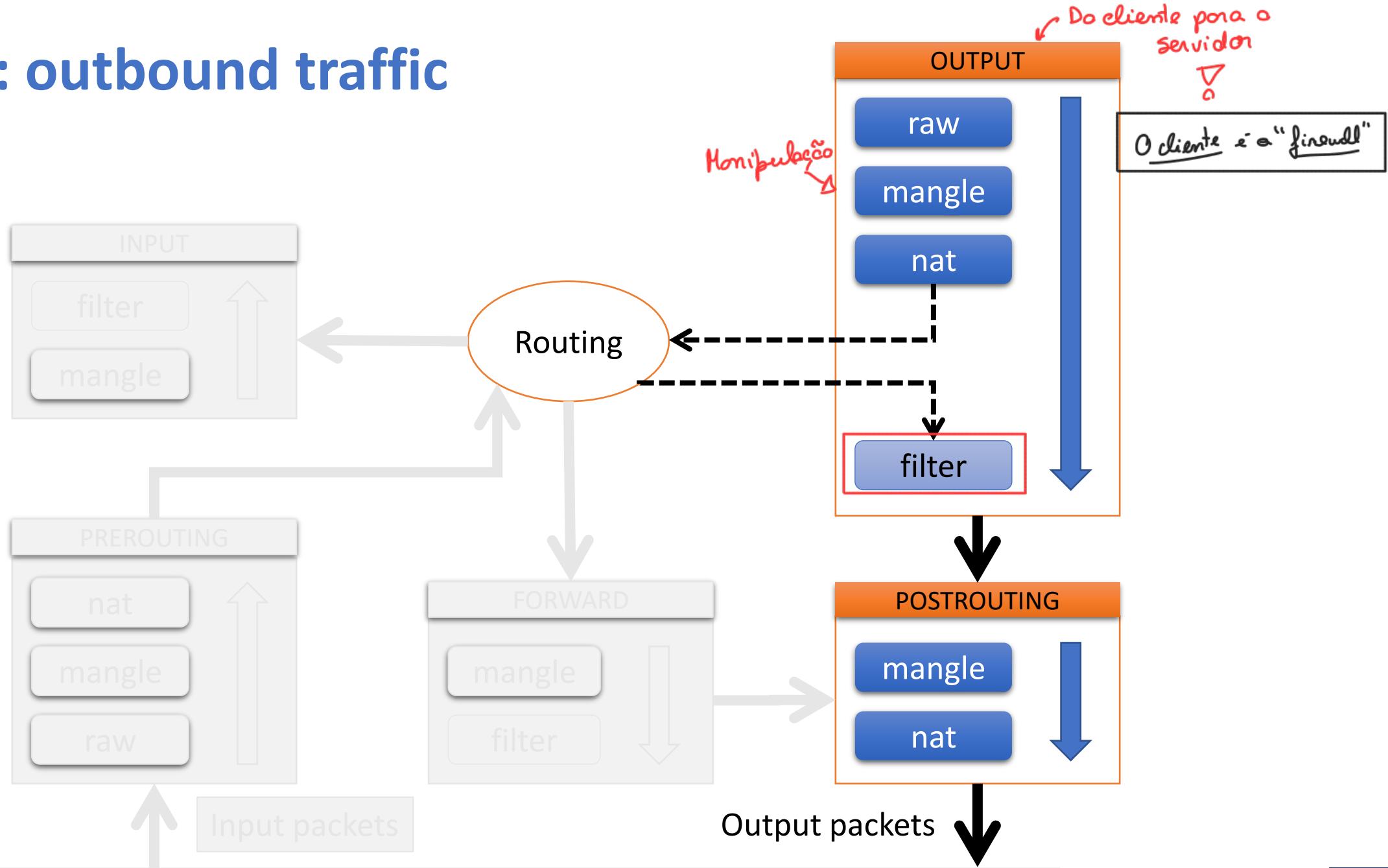
# Iptables: flows & chains



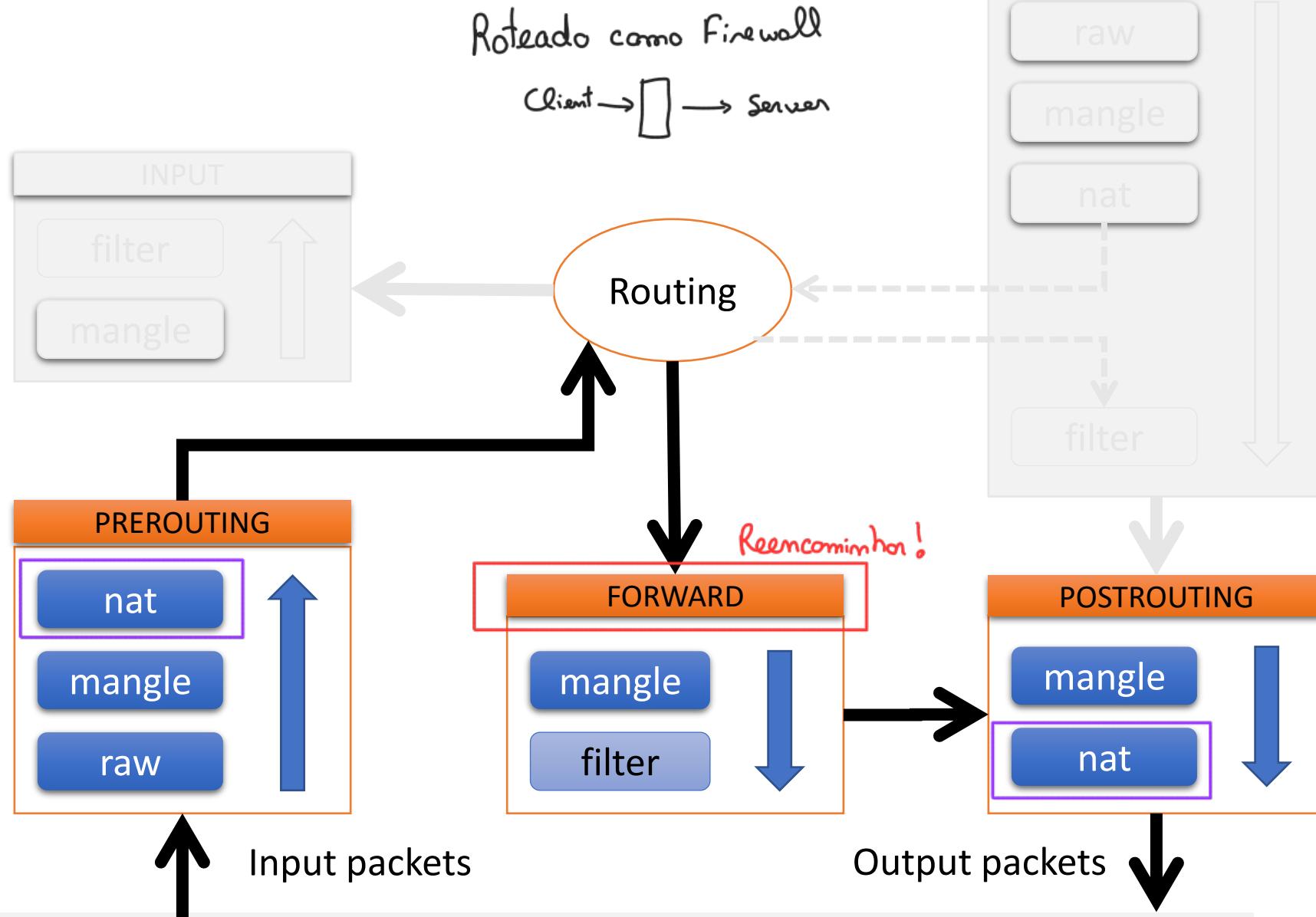
# Iptables: inbound traffic



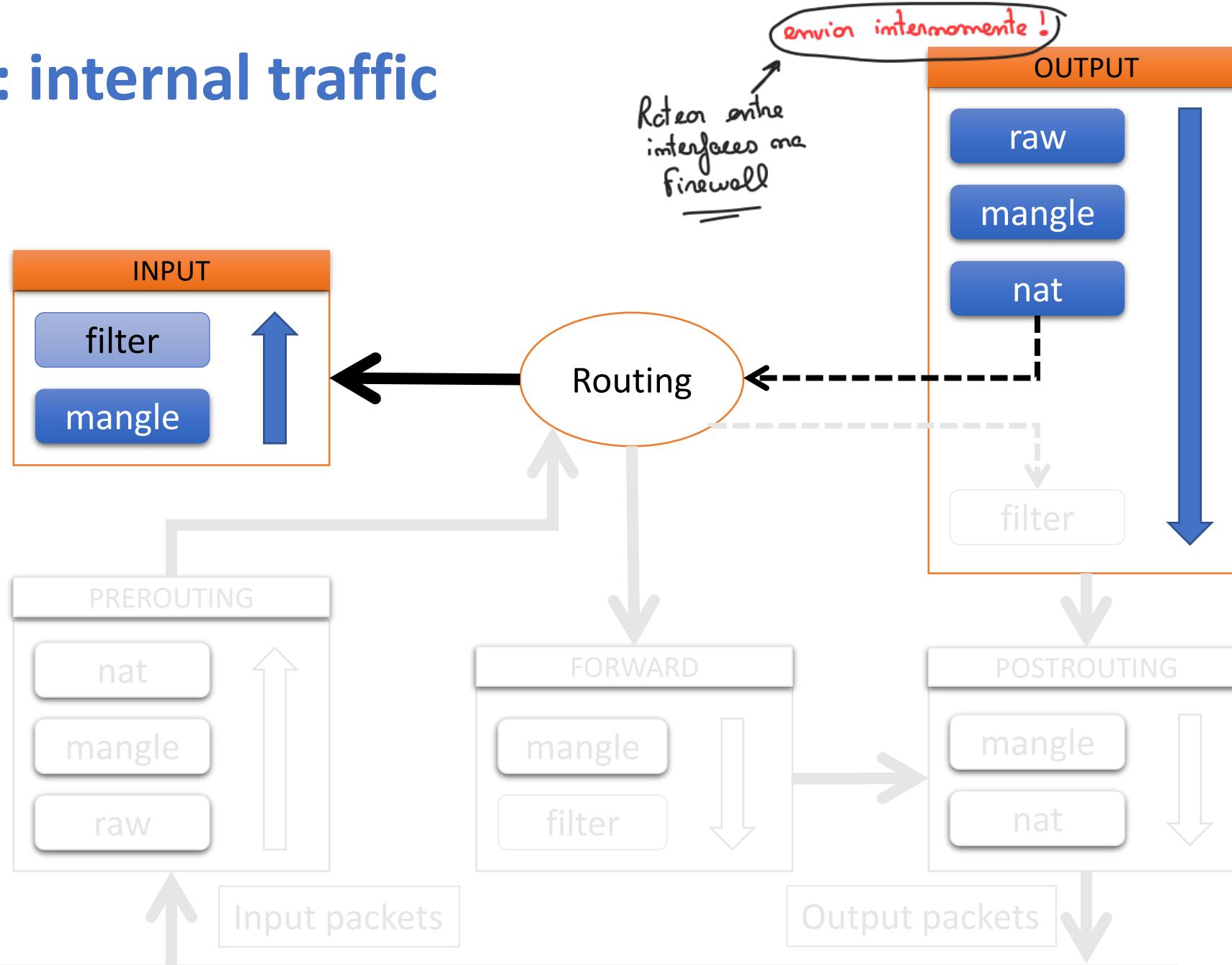
# Iptables: outbound traffic



# Iptables: en route traffic



# Iptables: internal traffic



# Iptables: decisions (or verdicts)

- Basic

- ACCEPT *Não nego de filter*
  - Let the package continue
- DROP
  - Discard the package
- CONTINUE
  - Use decisions from other rules

- Reusable Decisions

- New chains *Como se fasssem funções!*
- Jump to a new chain
  - The name of the chain is the decision
- RETURN
  - Leave the current chain

- Other

- LOG *Monito pacotes*
- MARK
  - With internal label
  - Useful for making coherent decisions across different chains
- REJECT *Aviso quem enviou o pacote do erro...*
  - Rejection with error message
- SNAT, MASQUERADE
  - Source NAT (masquerading)
- DNAT, REDIRECT
  - Destination NAT (port forwarding)
- QUEUE *Para uma aplicação!*
  - Forward to applications

Usando o mangle  
↓  
x Marco o pacote

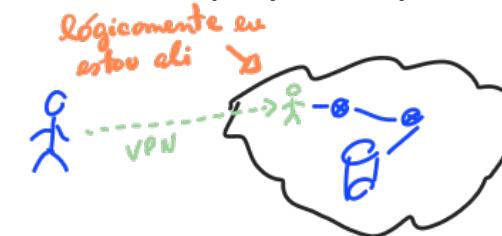
Processamento à entrada

Processamento à saída x se estiver marcado  
modifico

# VPN (Virtual Private Network)

Permite estender (logicamente) o perímetro de segurança!

- A VPN is a technology that extends a security perimeter
  - It allows traffic from a security perimeter to extend to hosts or networks physically far from it
  - The traffic in the VPN must be cryptographically protected



- In a nutshell, it combines two technologies

- IP or TCP/UDP tunneling
  - Over IP, UDP, TCP, or an applicational protocol
- Secure tunneling

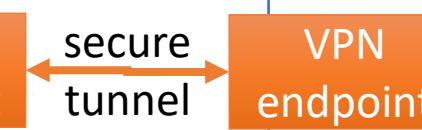
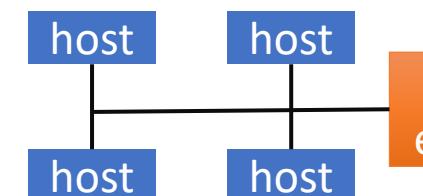
- Confidentiality & integrity control

## VPN types

- Host-to-Host
- Host-to-Net
- Net-to-Net

Poderemos fazer  
com o SSH

Network extended by a VPN



Network extended by a VPN

Network extended by a VPN

# Why do we need a VPN?

- To expose critical assets of a protected perimeter only to authorized people working geographically far from it
  - Work from home ← VPN da UA
  - Roaming personnel
- To guarantee that remote interactions with the protected perimeter do not reveal useful information for an attacker
  - Traffic eavesdropping is possible but useless, because it is encrypted !
- To protect from malicious network providers
  - These can do all sorts of traffic manipulation, since they intercept the entire communication
  - By using a VPN, a user primarily uses network services from the protected perimeter
    - Example: DNS resolvers
  - **People should always use a VPN when using public networks!**

## VPN examples: SSH

- An SSH session allows to define port-oriented tunnels
  - The SSH client can expose ports that represent ports in the SSH server host/network
  - The SSH server can expose ports that represent ports in the SSH client host/network
- A single SSH session can multiplex several tunnels
  - But these must be explicitly defined
- SSH tunnels are very convenient for several applications/scenarios
  - For implementing a secure FTP session, which uses two TCP streams
    - Control and data
  - For running X11 graphical applications on the server host that display on the client host
  - For remote accessing specific services in a protected network
    - Without exposing other network services

# Firewall example: OpenVPN

- OpenVPN implements a network-based host-to-net VPN
  - All traffic from a client to a given network is routed to it through the VPN
  - The VPN server represents the VPN client in that network
  - The VPN server extends the security perimeter of the network it serves
- OpenVPN types
  - L2 (TAP): the server propagates L2 broadcast/multicast traffic to the client
  - L3 (TUN): the server propagates L3 (IP) unicast traffic to the client

VPN da UA

# Firewall example: IPSec

- IPSec is IP with some security-related modifications

- Extra headers
- Possible payload encryption

→ Secure (encrypted) communications! //

- Extra headers

- AH (Authentication Header)

- For adding source authentication & integrity control to IP packets (header and payload)

- ESP (Encapsulating Security Payload)

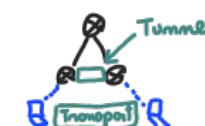
- For adding source authentication & integrity control and/or confidentiality to IP payloads

- IPSec modes

- Transport (rarely used)

both need the  
software which is difficult  
**Host-to-Host**

Solution  
(put software on the routers)



- Tunnel (used for host-to-net or net-to-net VPNs)

# Firewall example: IPSec

- An IPSec VPN is usually implemented with
  - IPSec in tunnel mode
  - ESP headers with source authentication, integrity control and confidentiality
- Original IP packets routed through the VPN are
  - Totally encrypted (both header and payload) on ingress
    - And decrypted and validated on egress
  - Become the payload of a VPN IP packet
  - The VPN IP packet has a proper ESP header
- But because of NAT, IPSec is tunneled over UDP
  - An extra **UDP header** allows NAT multiplexing

