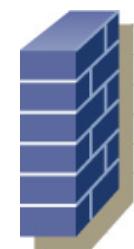
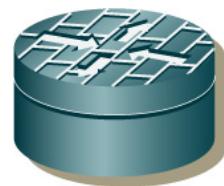


Firewalls

CCNA Security - Ch. 4: *Implementing Firewall Technologies*

Objectivos:

1. Descrever/configurar/depurar listas de controlo de acesso (ACLs):
 - Standard, estendidas, reflexivas, temporais, complexas e dinâmicas.
2. Mitigação de ataques comuns com ACLs.
3. Descrever os diversos tipos de *firewalls*, sua aplicação e impacto nas políticas de segurança da rede.
4. Descrever/configurar/depurar técnicas de controlo de acesso contextualizado (CBAC)
5. Descrever/configurar/depurar técnicas de controlo de acesso zonais (ZBPF)
6. Programação de firewalls com recurso ao SDM (ver capítulo SDM).



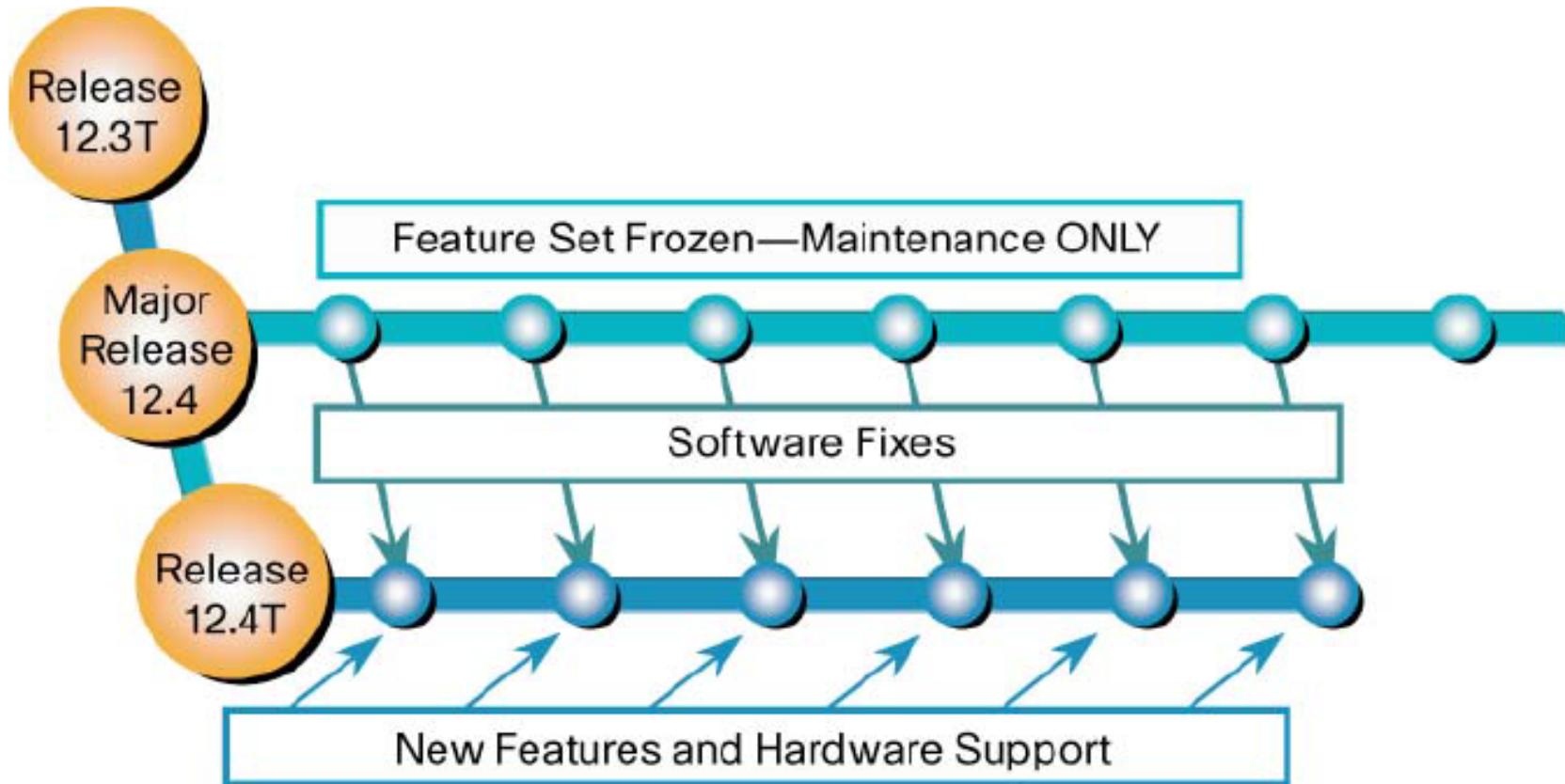
Tópicos

1. ACLs
2. Standard ACLs
3. Extended ACLs
4. Reflexive ACLs
5. Dynamic ACLs
6. Time-based ACLs
7. Troubleshooting de ACLs
8. Mitigação de ataques com ACLs
9. Tecnologias de Firewalls
10. Firewalls na arquitectura da rede
11. Content-based Access Control (CBAC)
12. Zone-based Policy Firewall (ZPF=ZBF=ZFW)

Lab Setup

DEIS

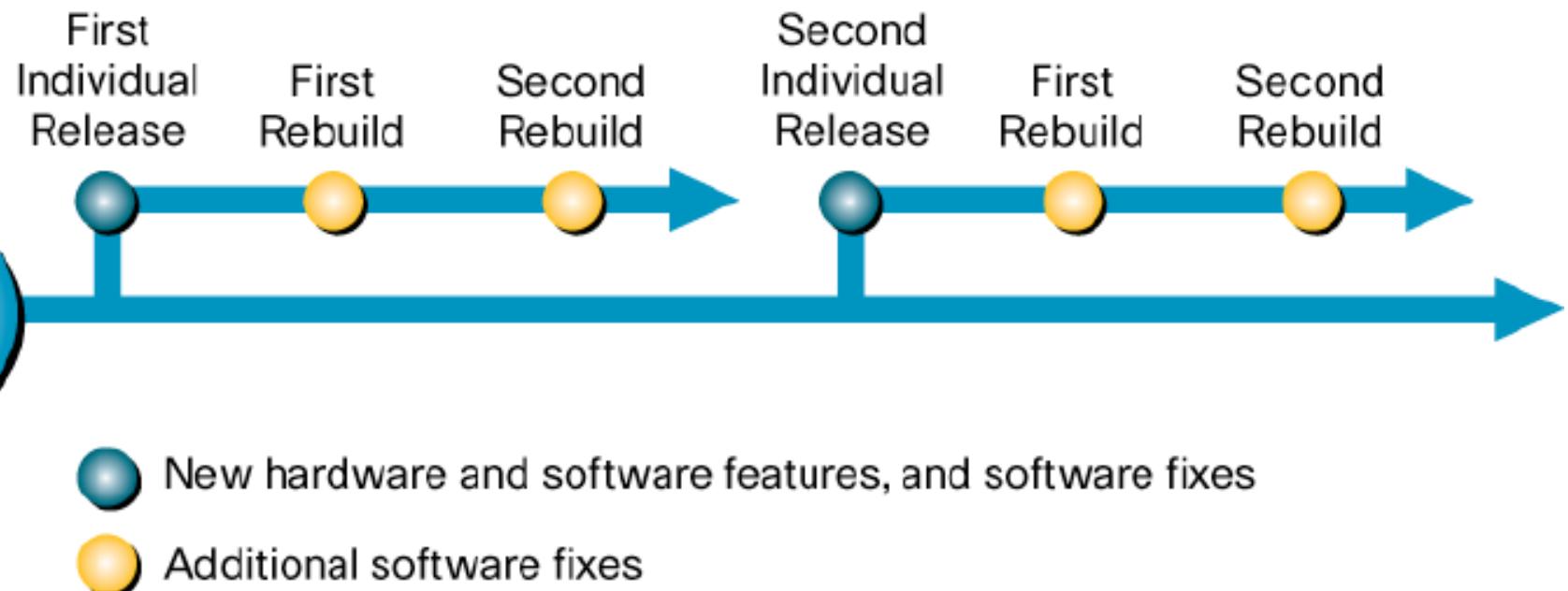
IOS 12.4T: Cisco IOS Packaging for Cisco Routers



Note: Technology releases are those Cisco IOS Software releases that introduce new features, functionality, and hardware support.

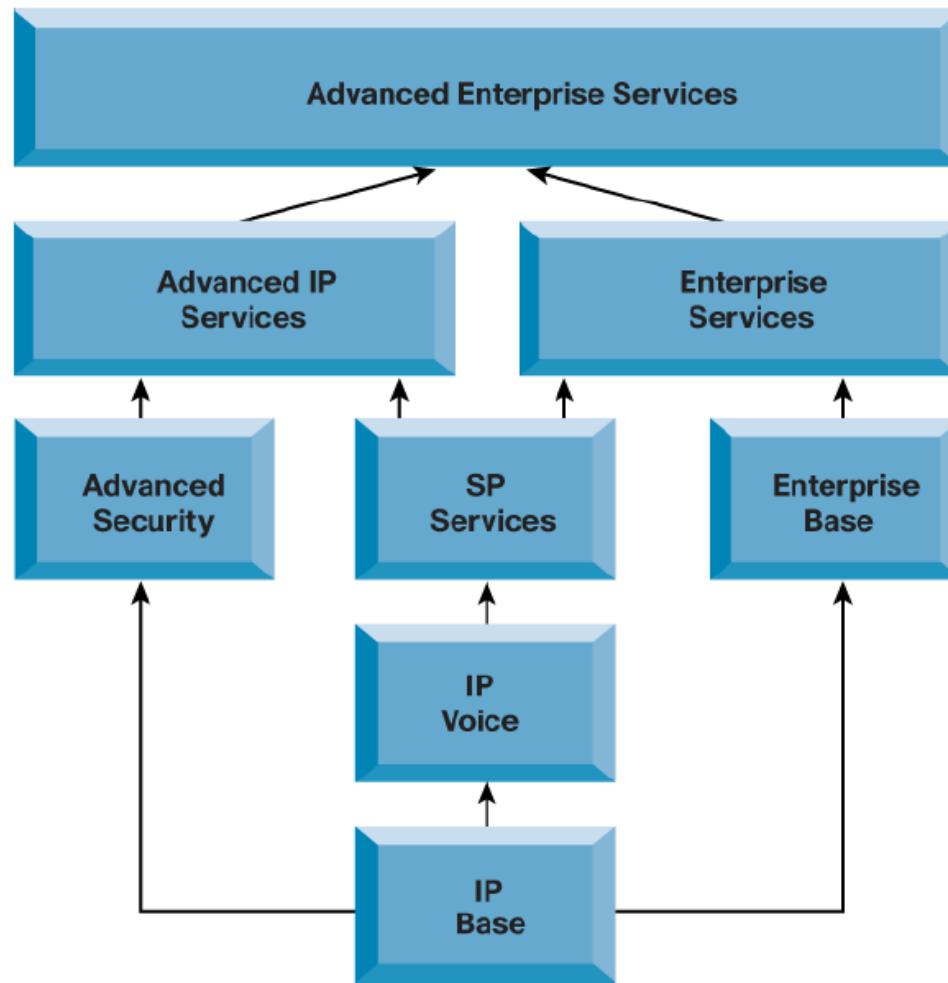


IOS 12.4T: Cisco IOS Packaging for Cisco Routers

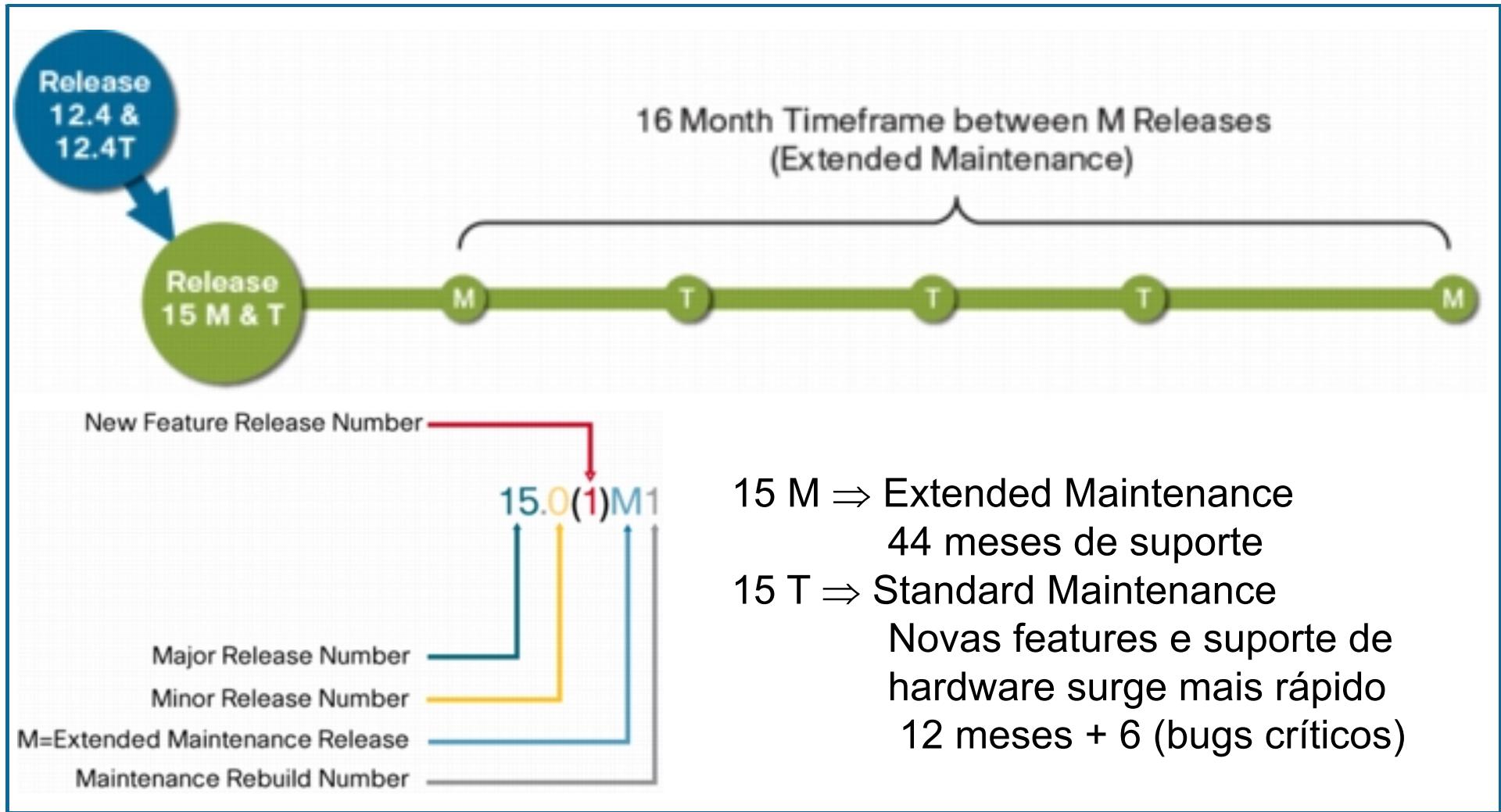


✓

IOS 12.4T: Cisco IOS Packaging for Cisco Routers

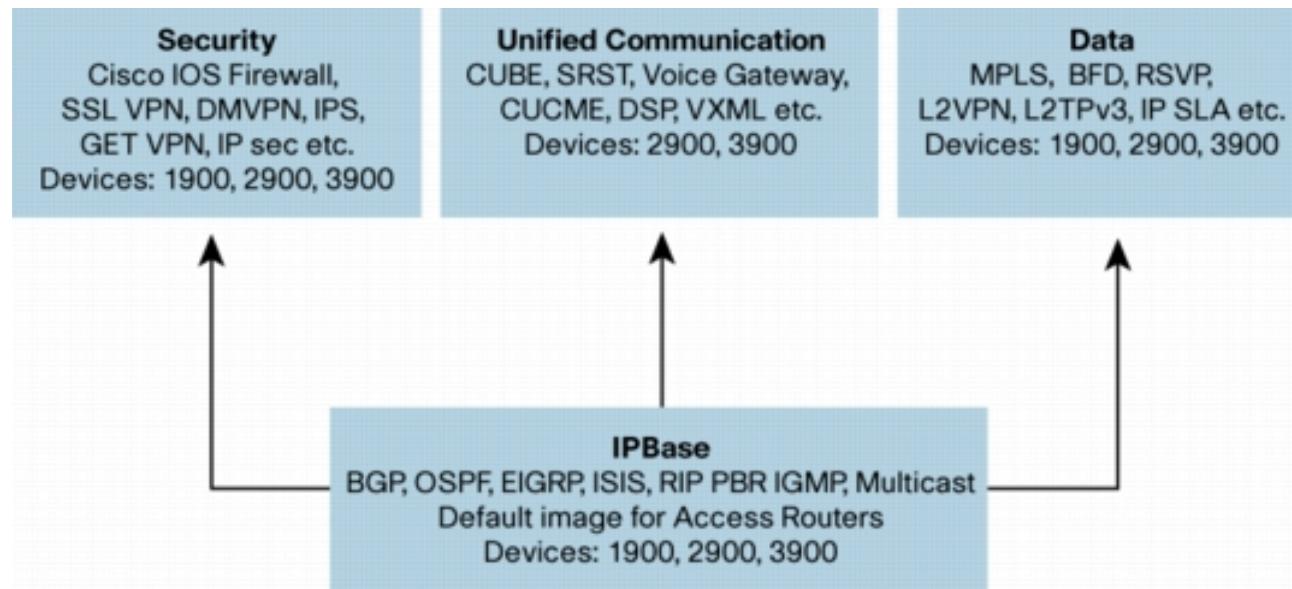


IOS 15: Cisco IOS Packaging for Cisco Routers

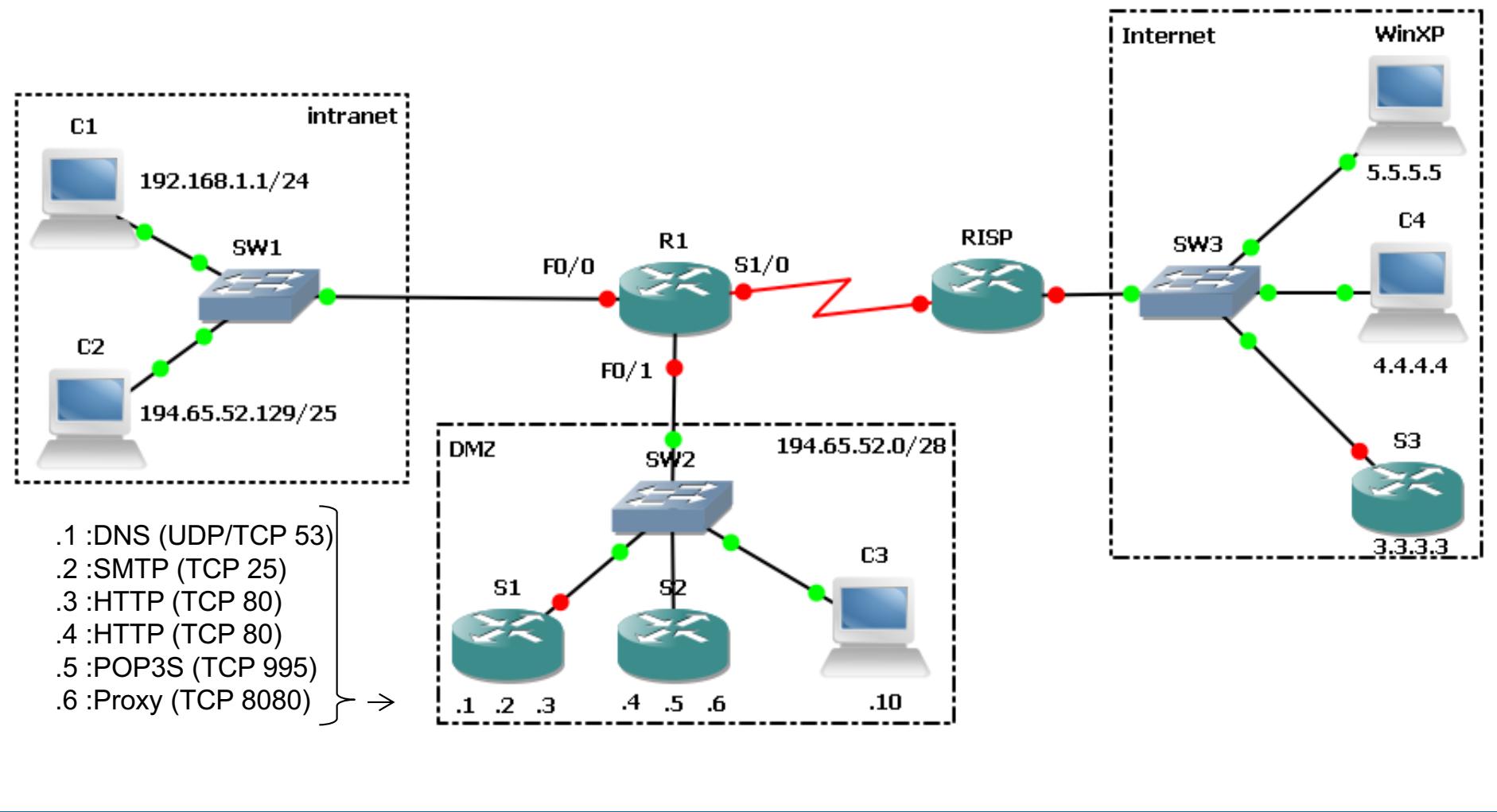


IOS 15: Cisco IOS Packaging for Cisco Routers

- Cisco IOS 15 (recordar a convenção das denominações)
 - As plataformas 1900, 2900 and 3900 Series Integrated Services Routers passam a ter uma única imagem universal
 - A ativação da variante específica (*Security, Unified Communications e Data*) é feita por licenciamento



Lab setup



Lab setup

- Como iniciar a rede
 - Topologia (firewall.zip):
 - Descomprimir para: Desktop > Cisco > GNS3 (IOS, Unix, PC Emulator) > Projects
 - Reset de configurações: Copiar ficheiros: configs-initial ⇒ configs
 - Duplo clique sobre o atalho “terminal-vpcs.exe”
 - Duplo clique sobre a topologia “topology.net”
 - Iniciar R1 e abrir consola; de seguida R-ISP
 - Nos exercícios em que tal for necessário, iniciar os servidores
 - Os servidores são routers IOS convertidos em hosts.
- Notas adicionais:
 - Os dois *routers* usados são Cisco 7206:
 - R1, R-ISP: c7200-advp�servicesk9-mz.151-4.M2.bin (26-Sep-11)
 - S1-S3: c2691-ipbase-mz.123-20.bin (imagem IOS muito leve mas antiga)
 - Interfaces IOS com múltiplos endereços (opção secondary no comando ip address)



ACLs (Introdução)

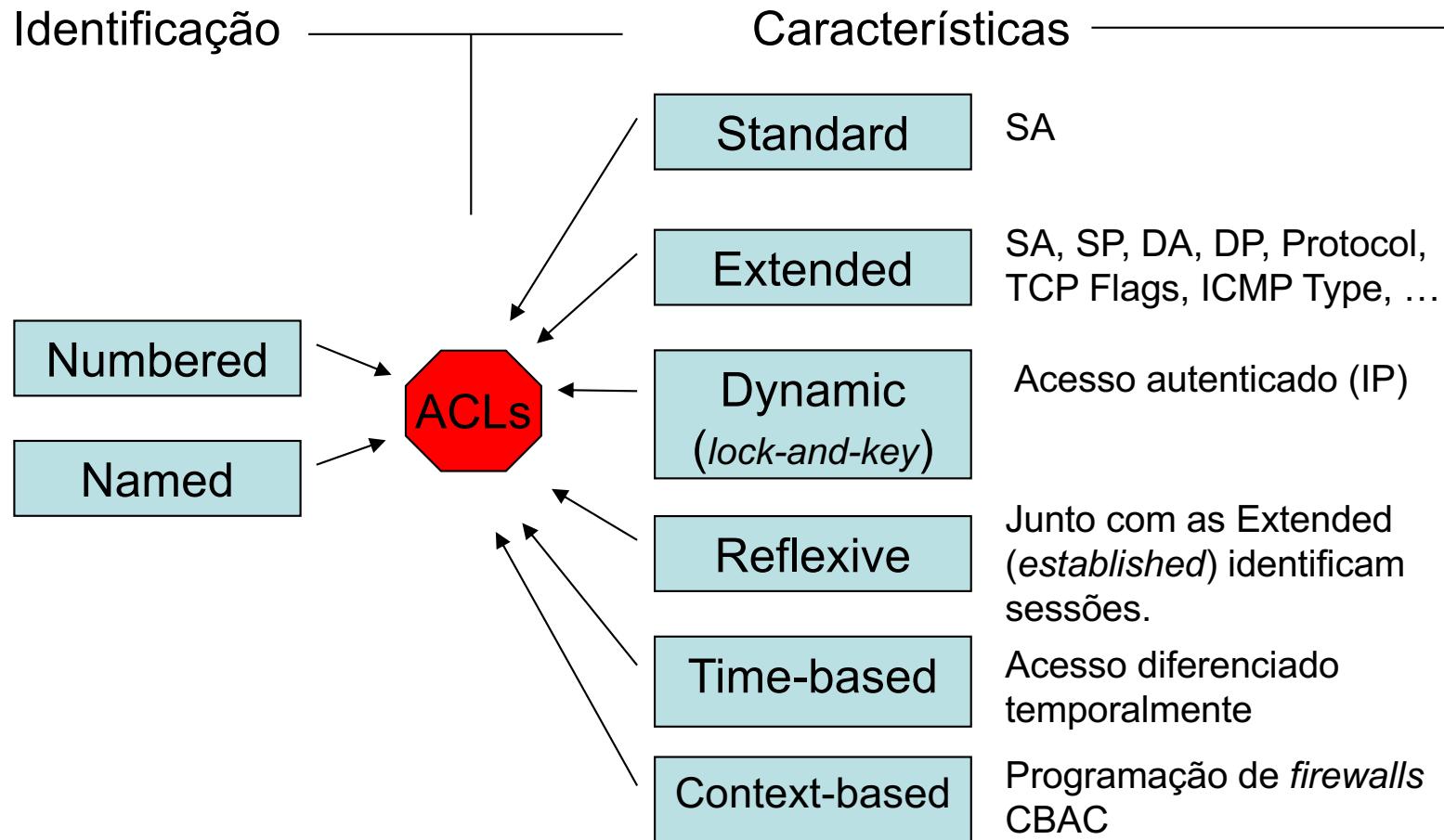
DEIS

Conceitos introdutórios

- O que são ACL?
 - São filtros que permitem identificar PDUs (i.e., quadros Ethernet, pacotes IPv4, pacotes IPv6, etc.) com base num conjunto de características (SA, DA, Flags, etc.) específico.
- Para que servem as ACL?
 - Condicionar fluxos de pacotes em trânsito (*firewalling*)
 - Condicionar acesso aos serviços do R1
 - Seleccionar fluxos para tratamento prioritário (QoS)
 - Seleccionar fluxos para encriptação e translação (NAT/PAT)
 - Seleccionar fluxos que desencadeiam ligações *dial-up* (WAN)
 - Controlar troca de rotas entre protocolos de encaminhamento
 - ...



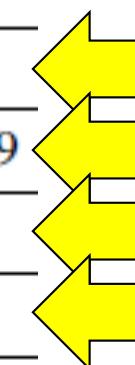
Conceitos introdutórios: tipos de ACL



Conceitos introdutórios

- Aplicação protocolar das ACL numeradas

Protocol	Range
IP	1–99, 1300–1999
Extended IP	100–199, 2000–2699
Ethernet type code	200–299
Ethernet address	700–799
Transparent bridging (protocol type)	200–299
Transparent bridging (vendor code)	700–799
Extended transparent bridging	1100–1199
DECnet and extended DECnet	300–399



Conceitos introdutórios

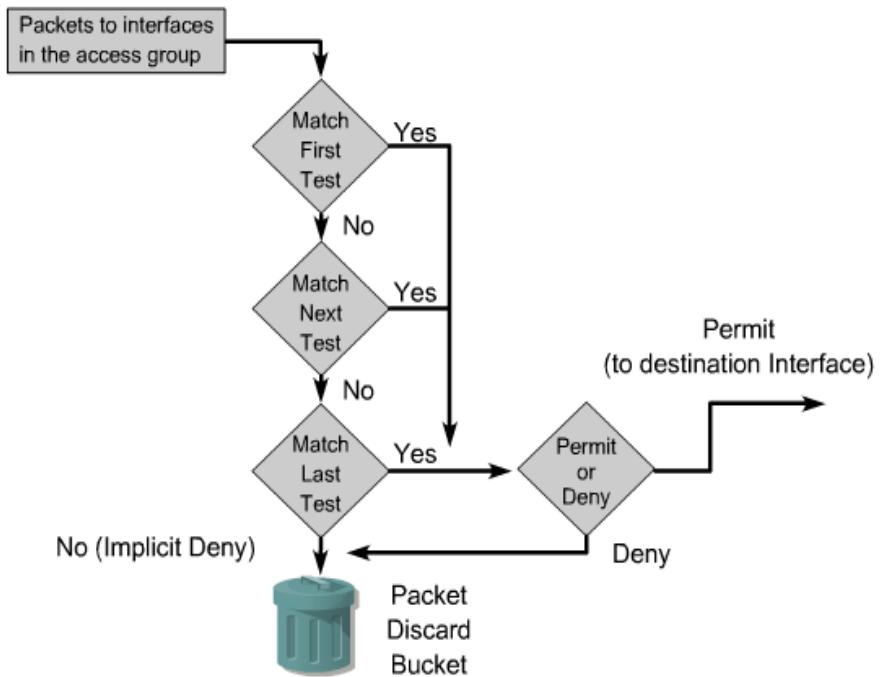
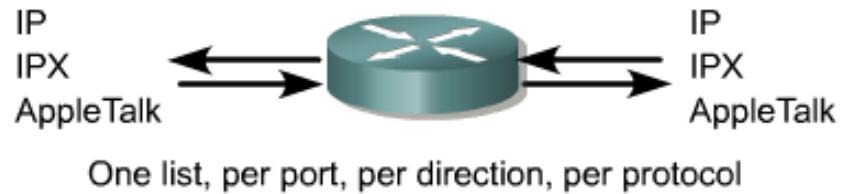
- Aplicação protocolar das ACL numeradas

Protocol	Range
XNS	400–499
Extended XNS	500–599
AppleTalk	600–699
Source-route bridging (protocol type)	200–299
Source-route bridging (vendor code)	700–799
IPX	800–899
Extended IPX	900–999
IPX SAP	1000–1099
Standard VINES	1–100
Extended VINES	101–200
Simple VINES	201–300



Conceitos introdutórios: regras de aplicação

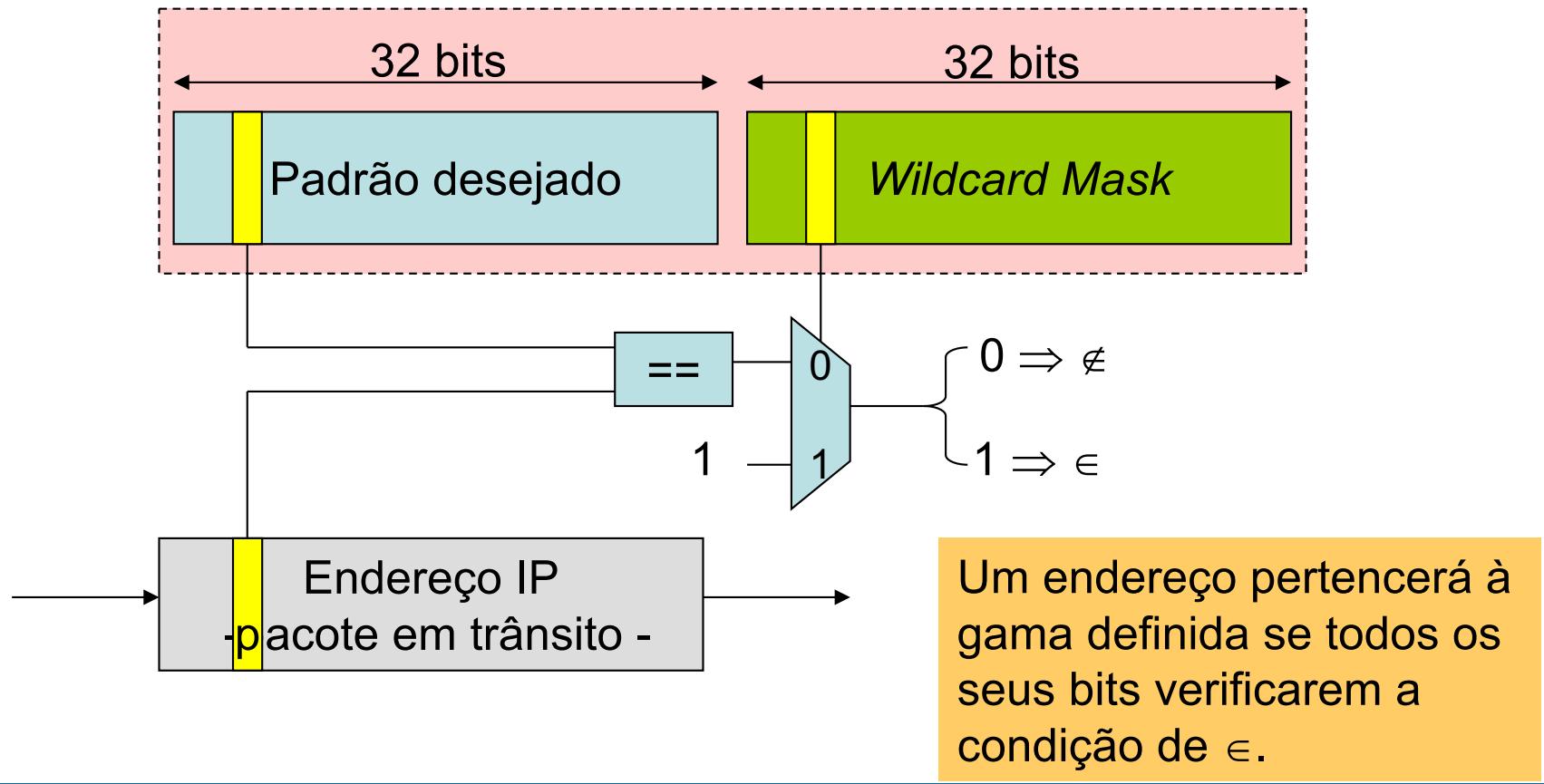
- Pode existir uma única ACL por protocolo (e.g. IP), interface e sentido (*in / out*)
- Uma ACL é constituída por uma sequência de testes
- Uma ACL deve conter testes que absorvam qualquer pacote. Este cuidado é assegurado por um teste implícito que o IOS considera no final: “**deny any**”
- Um pacote apenas é sujeito a um teste se não tiver verificado nenhum anterior



Conceitos introdutórios: regras de aplicação

- Representação de gamas de endereços

Gama de endereços IP



Conceitos introdutórios: exercício

- Como definir a gama de endereços 172.16.*.* ?

Padrão = 172.16.0.0

 1 0 1 0 1 1 0 0 0 0 0 1 0

Wildcard Mask = 0.0.255.255

 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

- Gama: 172.16.0.0 255.255.0.0
- Regra prática: No padrão substituir os bits '*' pelo valor binário '0' e na *wildcard mask* pelo valor binário '1'

- Caso especial: *.*.*.*

- Gama: 0.0.0.0 255.255.255.255 ≡ **any**

- Caso especial: a.b.c.d (ex.: 1.2.3.4)

- Gama: a.b.c.d 0.0.0.0 ≡ **host** a.b.c.d ≡ a.b.c.d



Conceitos introdutórios: fases de configuração

1. Definição

- Identificação do objectivo
- Selecção da posição
- Sequência de testes

Guideline	Benefit
Base your ACLs on the security policy of the organization.	This will ensure you implement organizational security guidelines.
Prepare a description of what you want your ACLs to do.	This will help you avoid inadvertently creating potential access problems.
Use a text editor to create, edit and save ACLs.	This will help you create a library of reusable ACLs.
Test your ACLs on a development network before implementing them on a production network.	This will help you avoid costly errors.

```
R1> enable
R1# configure terminal
R1(config)# access-list n {permit | deny } <Test1>
R1(config)# access-list n {permit | deny } <Test2>
R1(config)# ...
```

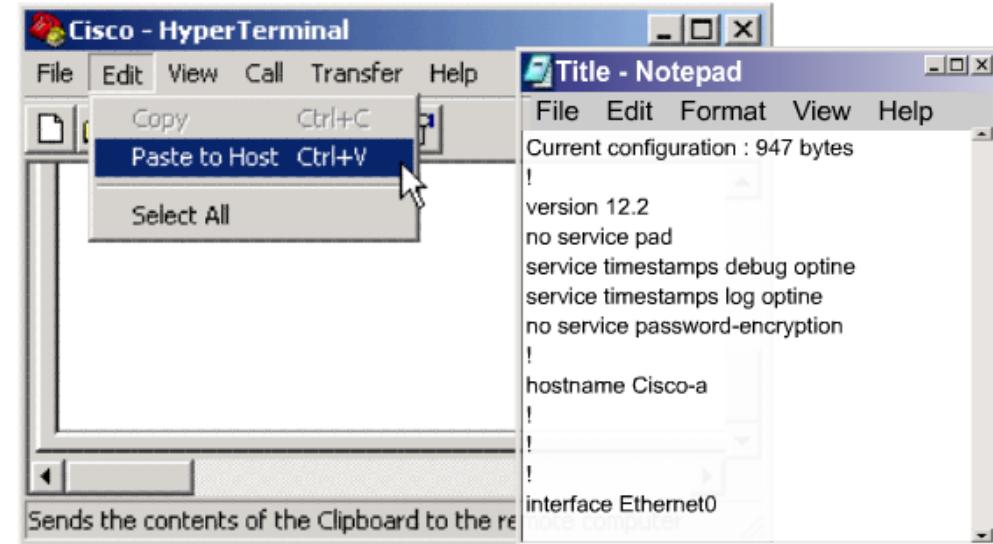
2. Aplicação

```
R1> enable
R1# configure terminal
R1(config)# interface <IF>
R1(config-if)# ip access-group <nº/nome> {in | out}
```

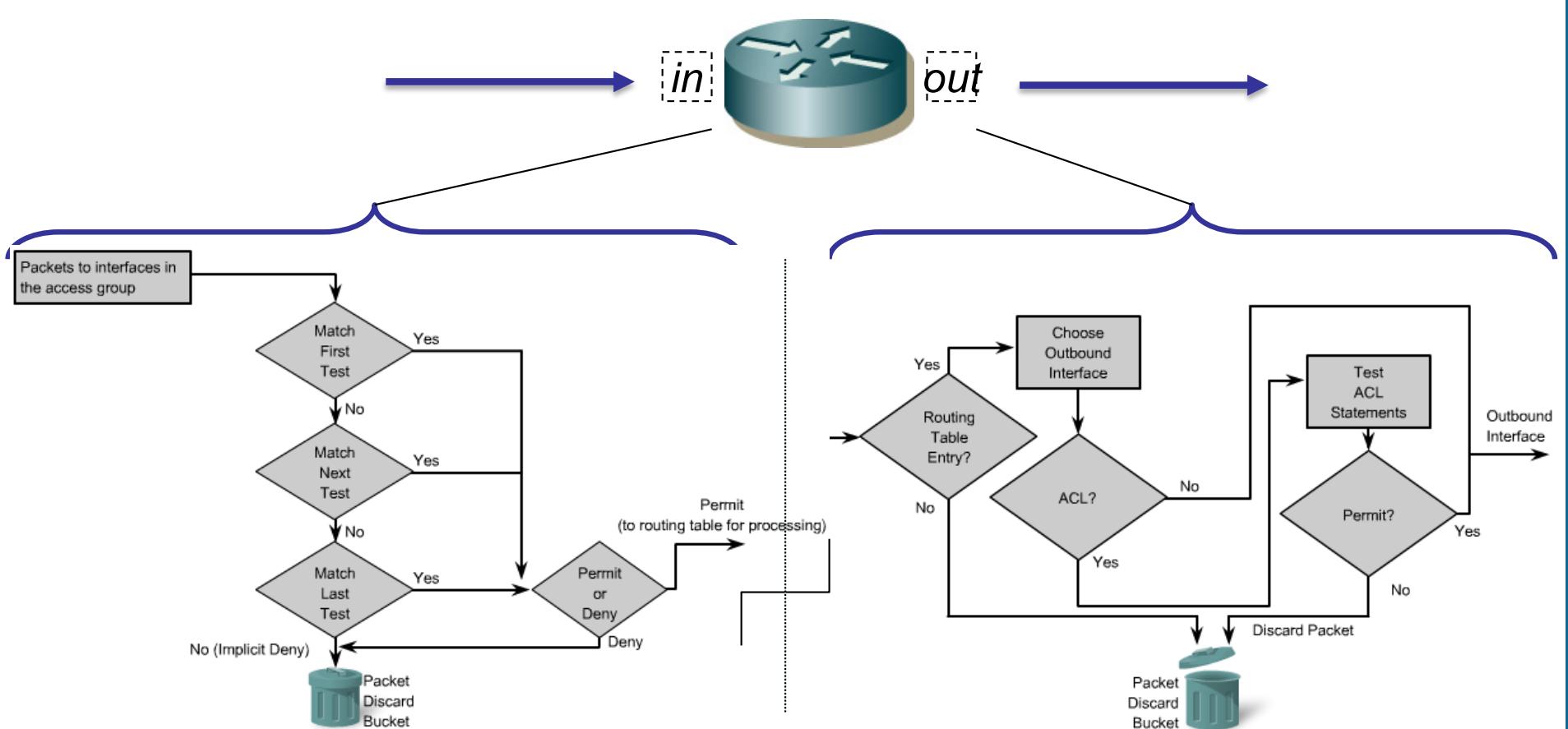


Conceitos introdutórios: fases de configuração

- A sequência exacta dos testes de uma ACL é determinante
- Para versões IOS anteriores à 12.3 a melhor estratégia para actualização de uma ACL era apagá-la no IOS, editá-la com um editor de texto e por *copy & paste* enviá-la novamente para o router.
- A partir da versão 12.3 é possível nas *named ACL* especificar explicitamente a sequência exacta de cada teste.
 - Qdo sequência não definida valor é incrementado de 10 em 10.



Conceitos introdutórios: fases de configuração



Nota: Tráfego produzido pelo *router* (protocolos de encaminhamento, CDP, etc.) não é abrangido por ACLs que processam tráfego de saída, só de entrada.



Conceitos introdutórios: fases de configuração

- Cisco Acces List Checker
 - <https://cway.cisco.com/tools/accesslist/>

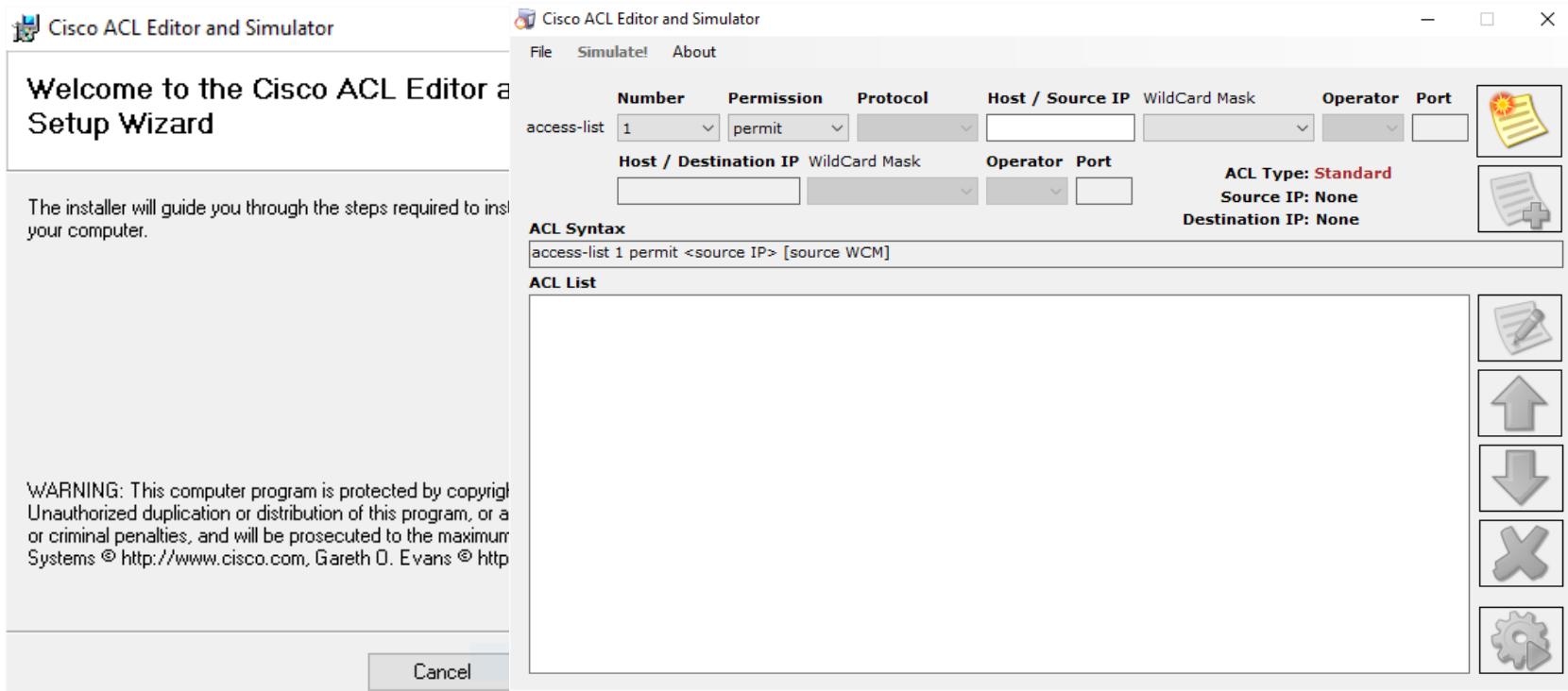
The screenshot shows a web browser window with the URL <https://cway.cisco.com/tools/accesslist/>. The page is titled "Tools Catalog / Access List" and features the Cisco logo. A blue banner at the top right asks for feedback: "Is the tool helpful? Let us know your feedback, click on 📡 in upper right." Below the banner, the title "Access List BETA" is displayed. A sub-header states: "The tool compares a SRC/DST IP+Port and checks to see if there is a matching entry in a Cisco IOS/NXOS access list. details ▾". A code snippet shows an Extended IP access list named "test-acl":

```
show ip access-list output
Extended IP access list test-acl
10 permit ip any range 1024 2048 host 192.168.1.2 eq www
11 remark ip any
20 permit ip 192.168.0.0/0 10.66.85.0 0.0.0.255
25 permit ip host 192.168.5.5 10.0.0.0 0.0.0.255
40 permit tcp host 10.66.86.1 lt 65530 any eq ftp-data
40 permit tcp any host 192.168.1.2 eq 80
30 permit ip 10.66.86.0 0.0.0.255 gt 1024 192.168.1.0 0.0.0.255
50 permit ip any any
```

At the bottom, it says "Source IP address 10.66.86.1" and "Source port 23001".

Conceitos introdutórios: fases de configuração

- Cisco ACL Editor and Simulator (<http://www.garethevans.info>)
 - Available on moodle



Standard ACLs

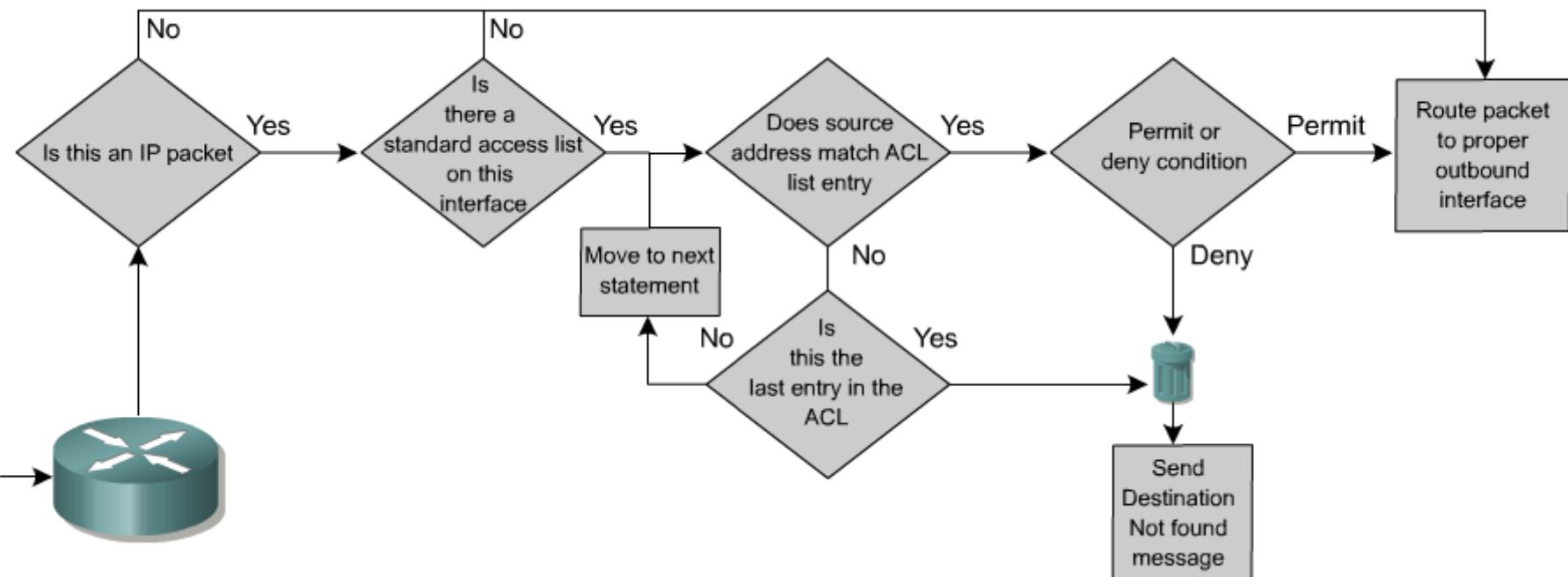
DEIS

ACL Standard

IPv4/v6: 1-99, 1300-1999

```
access-list access-list-number {deny | permit}  
source [source-wildcard] [log]
```

↑
Padrão de endereços fonte



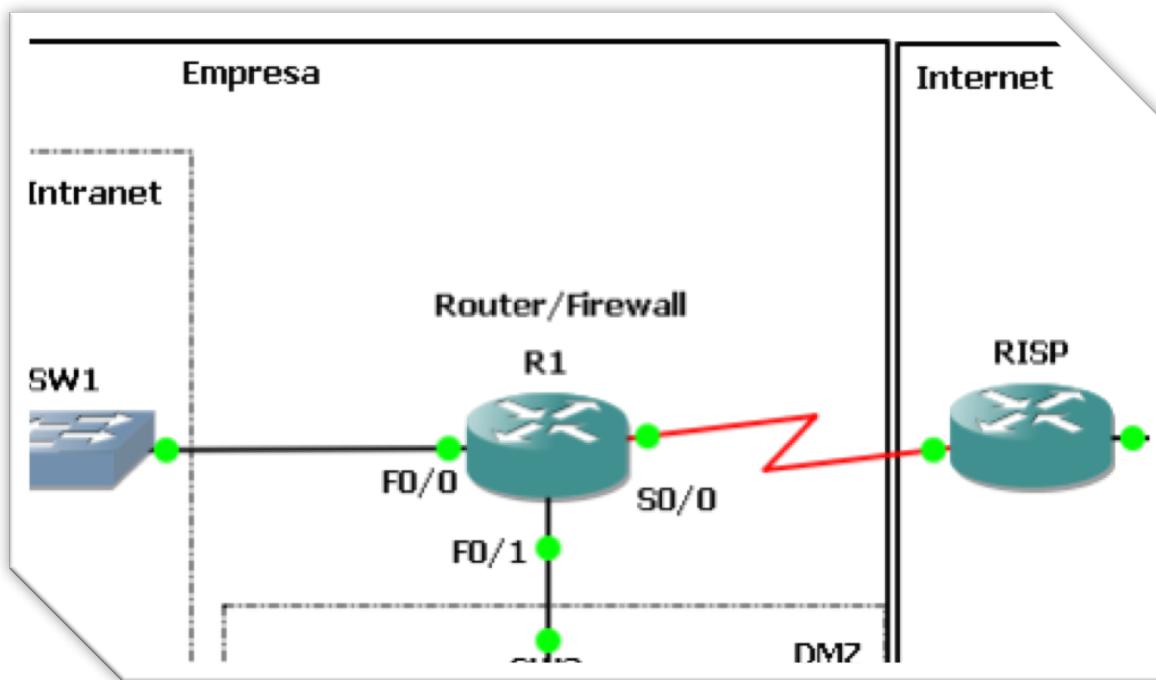
ACL Standard

Parameter	Description
access-list-number	Number of an ACL. This is a decimal number from 1 to 99, or 1300 to 1999 (for standard ACL).
deny	Denies access if the conditions are matched.
permit	Permits access if the conditions are matched.
remark	Add a remark about entries in an IP access list to make the list easier to understand and scan.
source	Number of the network or host from which the packet is being sent. There are two ways to specify the <i>source</i> : <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format.• Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.55.
source-wildcard	(Optional) Wildcard bits to be applied to the source. There are two ways to specify the <i>source-wildcard</i> : <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.• Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.55.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the logging console command.) The message includes the ACL number, whether the packet was permitted or denied, the source address, and the number of packets. The message is generated for the first packet that matches, and then at five-minute intervals, including the number of packets permitted or denied in the prior five-minute interval.



Exercício

- Assegure-se que a sua empresa não exporta tráfego para a Internet com endereços fonte privados:
 - 192.168.0.0/16; 172.16.0.0-172.31.255.255; 10.0.0.0/8



Exercício: solução

- Verificar que a empresa exporta endereços privados
 - Em R-ISP activar a depuração de pacotes IP

```
R-ISP# debug ip packet
IP packet debugging is on
```

Alternativa: usar o *wireshark*.

- A partir de C1 (que possui endereço privado) pingar C4

```
VPCS[1]# ping 4.4.4.4
```

- Explicar que o ping não é bem sucedido apenas porque o ISP não possui rotas para o espaço privado de endereçamento mas verificar (através do log de R-ISP) que o tráfego sai da empresa.

```
*Mar 1 00:11:10.395: IP: tableid=0, s=192.168.1.1 (Serial0/0), d=4.4.4.4
(FastEthernet0/0), routed via RIB
*Mar 1 00:11:10.399: IP: s=192.168.1.1 (Serial0/0), d=4.4.4.4
(FastEthernet0/0), g=4.4.4.4, len 48, forward
*Mar 1 00:11:10.403: IP: s=192.168.1.1 (Serial0/0), d=4.4.4.4
(FastEthernet0/0), len 48, encapsulation failed
```



Exercício: solução

- Primeira abordagem

- Empregar uma ACL que descarte tráfego de origem privada que abandone a empresa, dando livre trânsito ao restante.

```
R1(config)# access-list 1 deny 10.0.0.0 0.255.255.255
R1(config)# access-list 1 deny 172.16.0.0 0.15.255.255
R1(config)# access-list 1 deny 192.168.0.0 0.0.255.255
R1(config)# access-list 1 permit any
R1(config)# interface Serial 0/0
R1(config-if)# ip access-group 1 out
```

- Repetir a experiência anterior e depois consultar a ACL

```
R1-Firewall#show access-lists 1
Standard IP access list 1
    10 deny    10.0.0.0, wildcard bits 0.255.255.255
    20 deny    172.16.0.0, wildcard bits 0.15.255.255
    30 deny    192.168.0.0, wildcard bits 0.0.255.255 (10 matches)
    40 permit  any
```



Exercício: solução

- Segunda abordagem

- Programar R1 para apenas exportar tráfego com origem no espaço de endereçamento público da empresa
 - Mais simples e mais eficaz pois filtra igualmente uma percentagem grande de ataques de *spoofing* com origem na própria empresa.

```
R1(config)# no access-list 1
R1(config)# access-list 1 permit 194.65.52.0 0.0.0.255
R1(config)# access-list 1 deny any log
```

- Testar repetindo o procedimento atrás efectuado: C1#ping C4

```
R1-Firewall#
*Mar 1 01:00:06.583: %SEC-6-IPACCESSLOGNP: list 1 denied 0 0.0.0.0 ->
192.168.1.1, 1 packet
R1-Firewall#sh access-lists 1
Standard IP access list 1
    10 permit 194.65.52.0, wildcard bits 0.0.0.255
    20 deny    any log (5 matches)
```

Estratégia para, pontualmente, identificar sistemas prevaricadores da rede interna.
O argumento *log* **não deve** ser usado em produção: afecta o desempenho.

ACL Standard

- Normalmente devem situar-se próximo do destino do tráfego que controlam

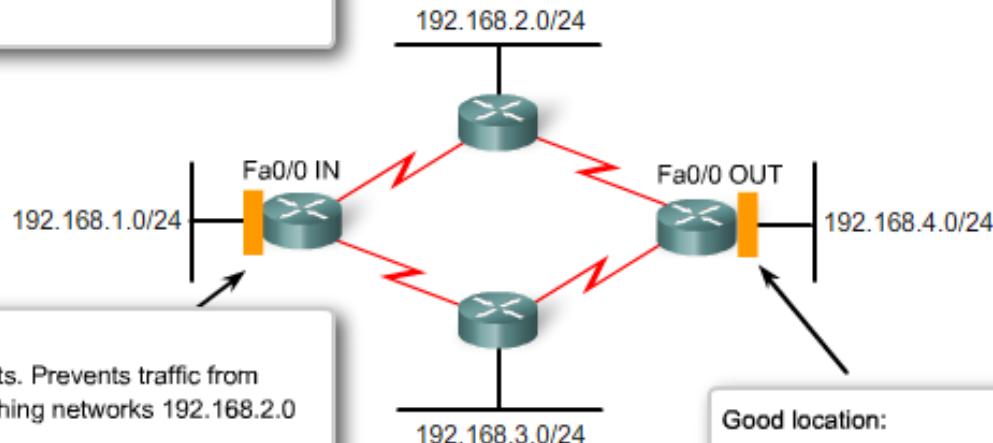
Requirements:

Prevent traffic from the 192.168.1.0 network from entering the 192.168.4.0 network. Allow 192.168.1.0 to reach other networks.

Bad Location:

Meets some of the requirements. Prevents traffic from 192.168.1.0 network from reaching networks 192.168.2.0 and 192.168.3.0.

Good location:
Meets all requirements.



```
ACL  
access-list 9 deny 192.168.1.0 0.0.0.255  
access-list 9 permit any
```



ACL Standard

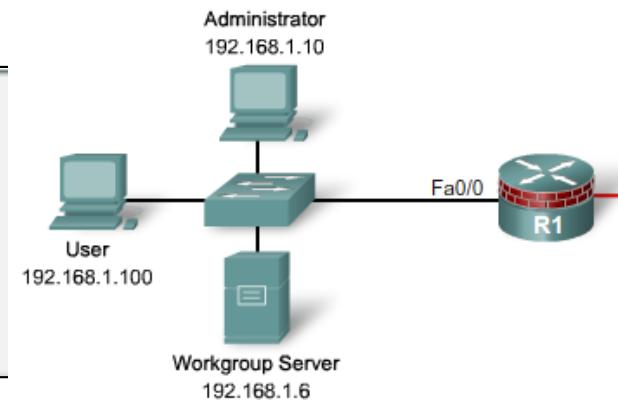
- Identificação com base em nomes (*named ACL*)

```
ip access-list [standard | extended] name  
Router(config-std-nacl) # [permit | deny | remark] {source  
[source-wildcard]} [log]
```

- Exemplo:

- Controlo de acesso aos serviços do router

```
R1(config)# ip access-list standard RESTRICT_VTY  
R1(config-std-nacl)# remark Permit only Admin host  
R1(config-std-nacl)# permit host 192.168.1.10  
R1(config-std-nacl)# exit  
R1(config)# line vty 0 4  
R1(config-line)# access-class RESTRICT_VTY in
```



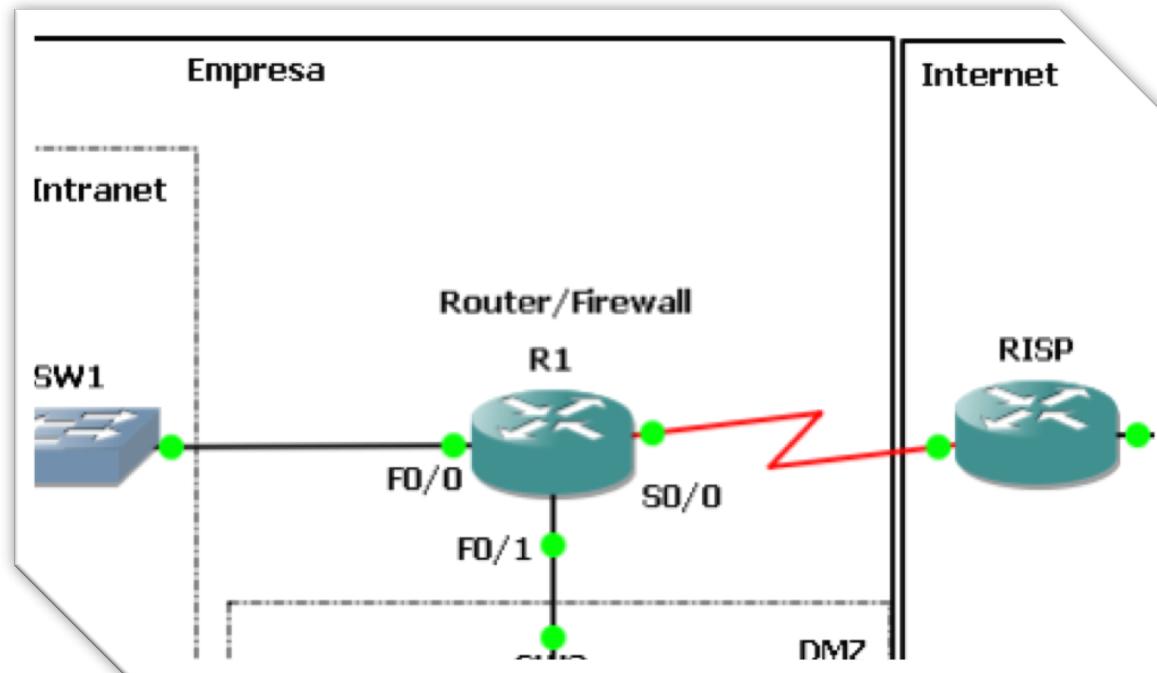
- Nota: Também é possível adicionar *remarks* nas *numbered ACL*

```
access-list number remark comment
```



Exercício

- Proteja a sua rede do *spoofing* típico vindo do exterior
 - Endereços públicos usados internamente; endereços *multicast*; endereços de *loopback*; 0.0.0.0; endereços privados
 - Recorra a *named ACLs* e teste a solução



Exercício: solução

- Exercício

```
conf term
    ip access-list standard anti-spoofing-acl
        remark Filtro Anti-spoofing
        deny 194.65.52.0 0.0.0.255
        deny 224.0.0.0 15.255.255.255
        deny 127.0.0.0 0.255.255.255
        deny host 0.0.0.0
        deny 10.0.0.0 0.255.255.255
        deny 172.16.0.0 0.15.255.255
        deny 192.168.0.0 0.0.255.255
    permit any

interface s0/0
    ip access-group anti-spoofing-acl in
end
```



Exercício: solução

- Activar o *debug* em R-ISP

```
R-ISP# debug ip packet  
IP packet debugging is on
```

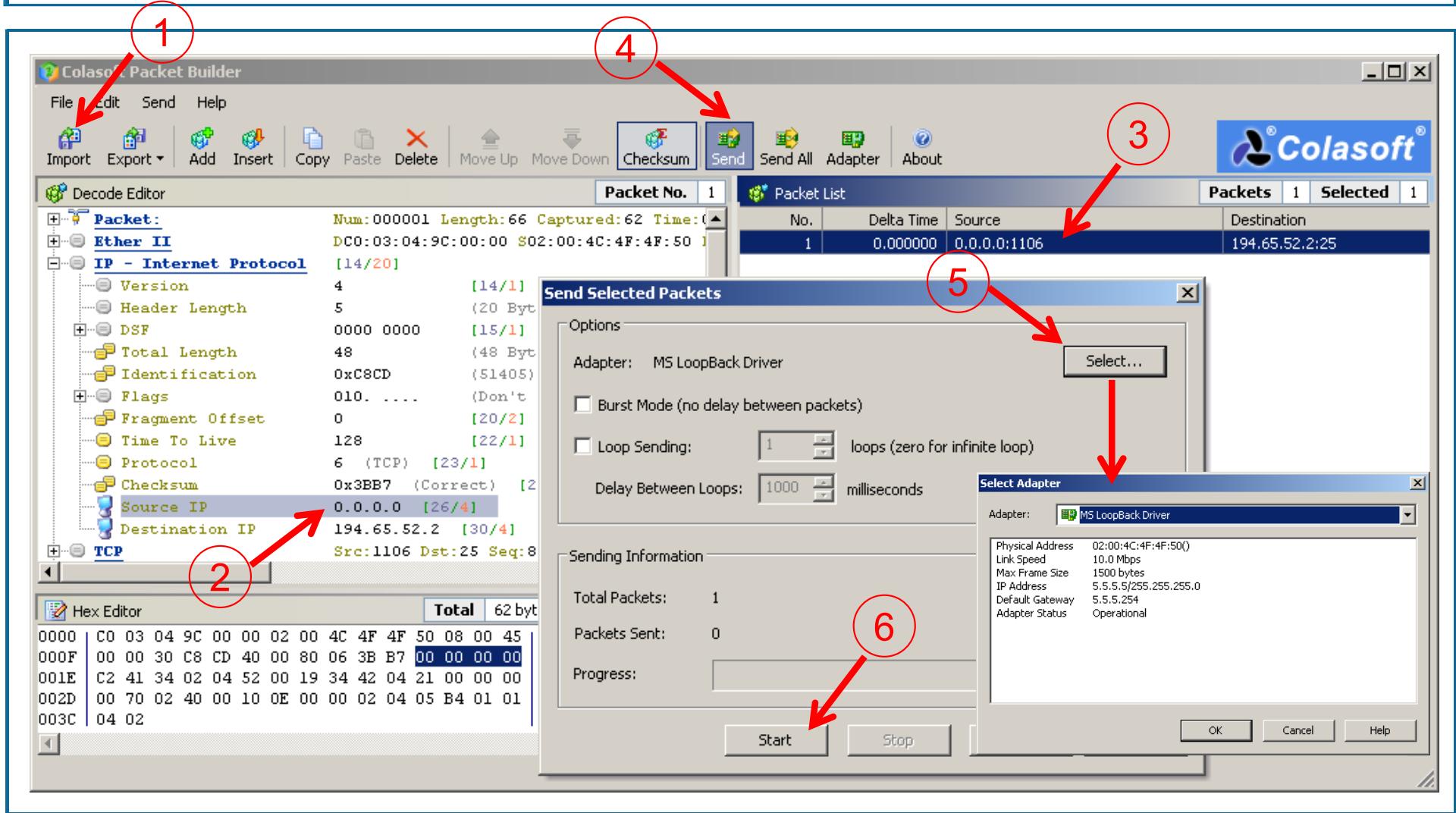
Alternativa: usar o wireshark.

- Enviar pacote *spoofed*
 - Pacote TCP dirigido ao *mail server* empresarial com SA=0.0.0.0
 - WinXP: Desktop > Security > ColapSoft Packet Builder
 - Ficheiro: WinXP-to-MailServer-SpoofedPacket.cscpkt
 - Em: Desktop > Cisco > GNS3 (IOS, Unix, PC Emulator) > Projects > Firewall
- Analisar a actividade de log em R-ISP

```
R1-Firewall#  
*Mar 1 04:36:27.382: IP: s=0.0.0.0 (Serial0/0),  
d=194.65.52.2, len 48, access denied
```



Exercício: solução

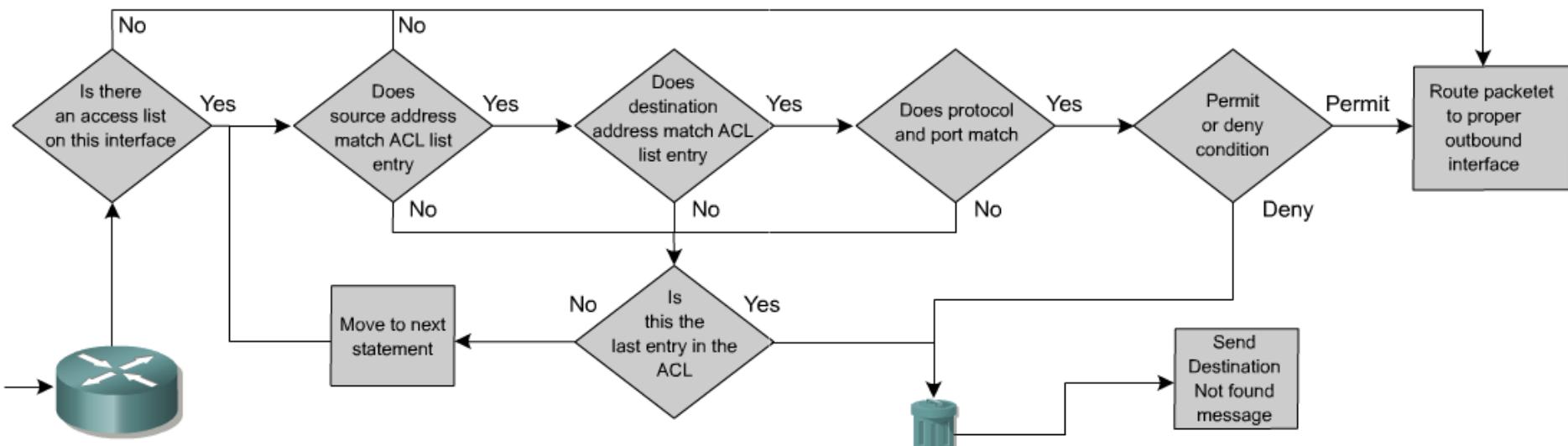


Extended ACLs

DEIS

ACL Extended

IPv4/v6: 100-199, 2000-2699
Porto fonte
access-list access-list-number {deny | permit} protocol source [source-wildcard] [operator operand] destination [destination-wildcard] [operator operand] [established] [log]
Porto destino



ACL Extended

Parameter	Description
<i>access-list-number</i>	Identifies the access list using a number in the range 100 to 199 (for an extended IP ACL) and 2000 to 2699 (expanded IP ACLs).
deny	Denies access if the conditions are matched.
permit	Permits access if the conditions are matched.
remark	Indicates whether this entry allows or blocks the specified address. Could also be used to enter a remark.
protocol	Name or number of an Internet protocol. Common keywords include icmp , ip , tcp , or udp . <u>To match any Internet protocol (including ICMP, TCP, and UDP) use the ip keyword.</u>
source	Number of the network or host from which the packet is being sent.
source-wildcard	Wildcard bits to be applied to source.
destination	Number of the network or host to which the packet is being sent.
destination-wildcard	Wildcard bits to be applied to the destination.
operator	(Optional) Compares source or destination ports. Possible operands include lt (less than), gt (greater than), eq (equal), neq (not equal), and range (inclusive range).
port	(Optional) The decimal number or name of a TCP or UDP port.
established	(Optional) For the TCP protocol only: Indicates an established connection.

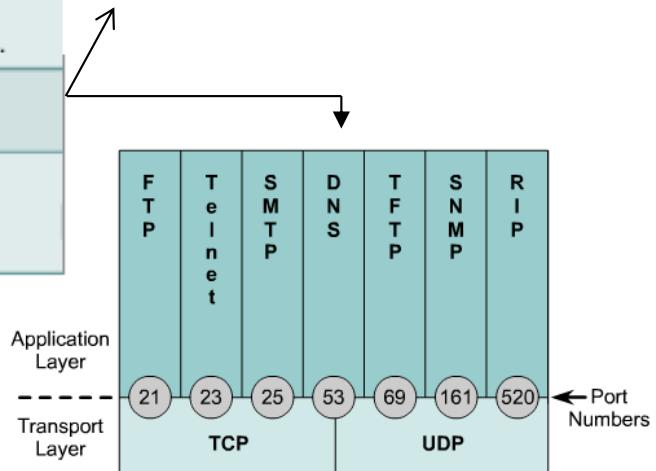


ACL Extended

Parameter	Description
<code>access-list-number</code>	Identifies the list using a number in the range 100 to 199.
<code>permit deny</code>	Indicates whether this entry allows or blocks the specified address.
<code>protocol</code>	The protocol, such as IP, TCP, UDP, ICMP, GRE, or IGRP.
<code>source and destination</code>	Identifies source and destination addresses.
<code>source-mask and destination-mask</code>	Wildcard mask; zeros indicate positions that must match, ones indicate do not care positions.
<code>operator operand</code>	lt, gt, eq, neq (less than, greater than, equal, not equal), and a port number.
<code>established</code>	Allows TCP traffic to pass if the packet uses an established connection (for example, has ACK bits set).

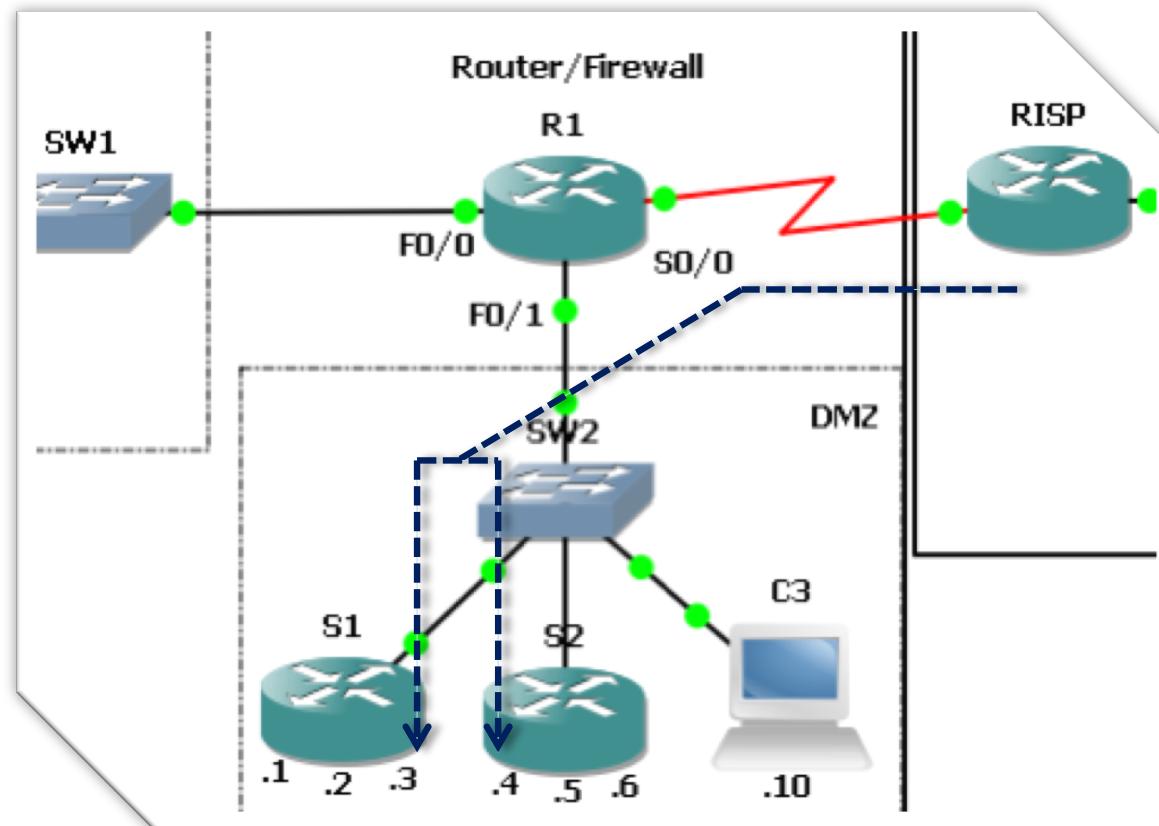
Mensagem TCP possui os bits ACK ou RST activos (todos os segmentos excepto o 1º) Útil quando queremos identificar segmentos TCP pertencentes a uma sessão aberta no sentido contrário.
Esta keyword foi introduzida em 1995.

O operador *range a b* pode ser usado para especificar intervalos de portos (útil por exemplo para FTP)



Exercício

- Assegurar que pedidos HTTP externos são limitados aos servidores web empresariais (194.65.52.3/.4)



Solução

- Verificar conectividade HTTP entre C4 e C3

```
VPCS[4]> ping 194.65.52.10 -3 -l 100 -p 80
Connect 80@194.65.52.10 seq=1 ttl=62 time=73.000 ms
SendData 80@194.65.52.10 seq=1 ttl=62 time=48.000 ms
Close    80@194.65.52.10 seq=1 ttl=62 time=37.000 ms
```

- Definição e aplicação da lista de controlo de acesso

```
access-list 100 permit tcp any host 194.65.52.3 eq www
access-list 100 permit tcp any host 194.65.52.4 eq www
access-list 100 deny tcp any any eq www
access-list 100 permit ip any any

interface Serial 0/0
  ip access-group 100 in
end
```



Solução

- Activar o *debug* em R1

```
R1-Firewall# debug ip packet
IP packet debugging is on
```

Alternativa: usar o wireshark.

- Verificar a perda de conectividade HTTP entre C4 e C3

```
VPCS[4]> ping 194.65.52.10 -3 -l 100 -p 80
Connect 80@194.65.52.10 timeout
```

- Nota: No decursos das experiências pode ser interessante inicializar os contadores de matches nas ACL

```
R1-Firewall#clear ip access-list counters {number | name}
```



Solução

- Acompanhar em R1

```
R1-Firewall#  
*Mar 1 06:13:15.238: IP: s=4.4.4.4 (Serial0/0), d=194.65.52.10, len  
140, access denied  
*Mar 1 06:13:15.242: IP: tableid=0, s=194.65.53.1 (local),  
d=4.4.4.4 (Serial0/0), routed via FIB  
*Mar 1 06:13:15.250: IP: s=194.65.53.1 (local), d=4.4.4.4  
(Serial0/0), len 56, sending  
R1-Firewall#  
*Mar 1 06:13:17.466: IP: s=4.4.4.4 (Serial0/0), d=194.65.52.10, len  
140, access denied  
  
R1-Firewall#sh access-lists 100  
Extended IP access list 100  
    10 permit tcp any host 194.65.52.3 eq www  
    20 permit tcp any host 194.65.52.4 eq www  
    30 deny tcp any any eq www (9 matches) ←  
    40 permit ip any any
```



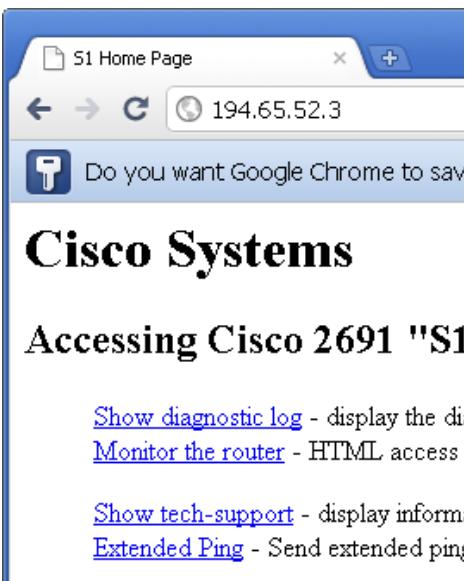
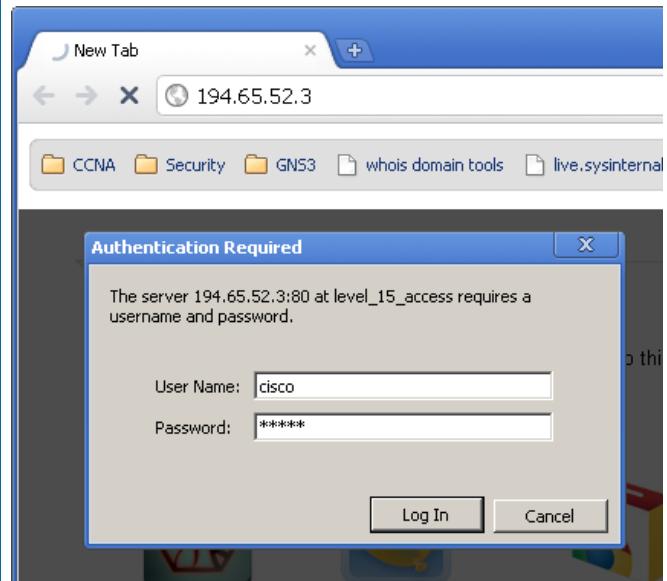
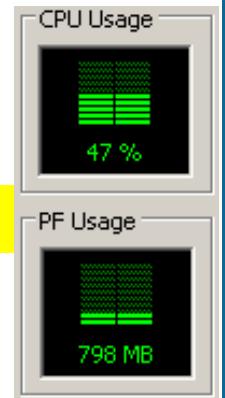
Solução

- Verificar conectividade HTTP entre WinXP e S1 (.3)

1. Iniciar o servidor S1 e lançar o servidor HTTP (+270 MB)

```
S1 (config) # ip http server
```

2. WinXP: Desktop > Chrome (+24 MB) > http://194.65.52.3
3. Entrar com as credenciais cisco/cisco



```
R1-Firewall#sh access-lists 100
```

Extended IP access list 100

10 permit tcp any host 194.65.52.3 eq www (50 matches) →

20 permit tcp any host 194.65.52.4 eq www

30 deny tcp any any eq www (9 matches)

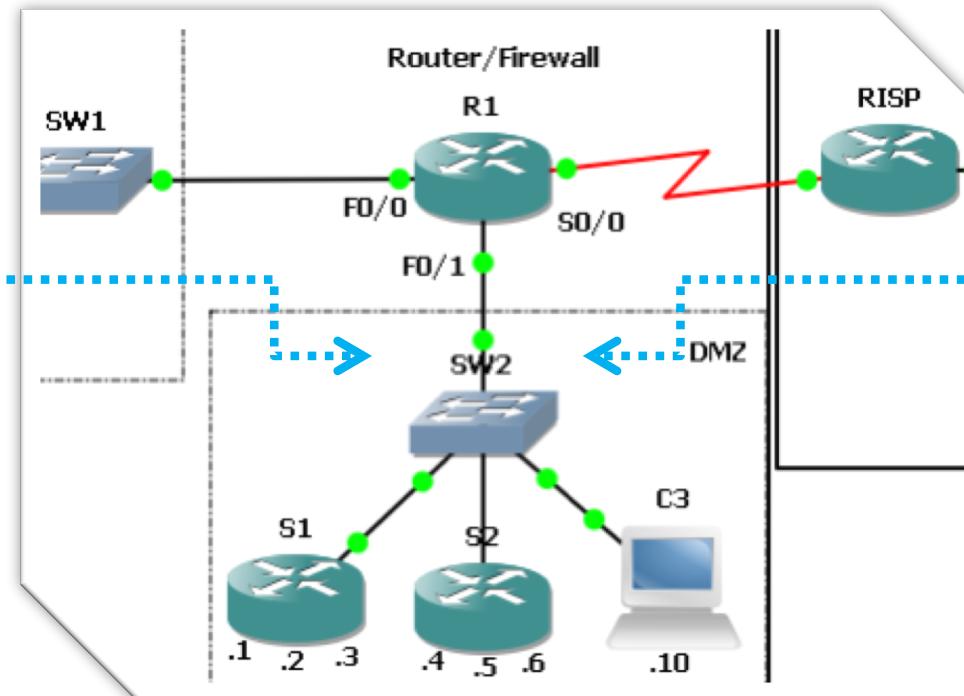
40 permit ip any any

4. Verificar perda de conectividade com S1 (.1)



Solução: verificação sistemática

- Depois de aplicadas as ACL pretendidas a sua efectiva utilização pode ser verificada a partir de diversos pontos de referência com um *port scanner* como o nmap.



Solução: verificação sistemática

- WinXP > *command prompt*

```
C:\Documents and Settings\Cisco>nmap --system-dns 194.65.52.3
Starting Nmap 5.21 ( http://nmap.org ) at 2012-04-03 09:34 GMT
Daylight Time
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
Nmap scan report for 194.65.52.3
Host is up (0.038s latency).

Not shown: 998 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 42.89 seconds
```



Solução: verificação sistemática

- Abordagem do *network mapper* (nmap)

No.	Time	Source	Destination	Protocol	Info
6	14.419000	5.5.5.5	194.65.52.3	ICMP	Echo (ping) request (id=0x57a7, seq(be/le)=0/0, ttl=38)
7	14.436000	5.5.5.5	194.65.52.3	TCP	37836 > 443 [SYN] Seq=583682051 Win=2048 Len=0 MSS=1460
8	14.448000	5.5.5.5	194.65.52.3	TCP	37836 > 80 [ACK] Seq=0 Ack=583682051 Win=4096 Len=0
9	14.492000	194.65.52.3	5.5.5.5	ICMP	Echo (ping) reply (id=0x57a7, seq(be/le)=0/0, ttl=254)
10	14.513000	5.5.5.5	194.65.52.3	ICMP	Timestamp request (id=0x44f2, seq(be/le)=0/0, ttl=36)
11	14.533000	194.65.52.3	5.5.5.5	TCP	443 > 37836 [RST, ACK] Seq=0 Ack=583682052 Win=0 Len=0
12	14.533000	194.65.52.3	5.5.5.5	TCP	80 > 37836 [RST] Seq=583682051 Win=0 Len=0
13	14.559000	194.65.52.3	5.5.5.5	ICMP	Timestamp reply (id=0x44f2, seq(be/le)=0/0, ttl=254)
14	14.624000	5.5.5.5	194.65.52.3	TCP	37836 > 111 [SYN] Seq=2796822245 Win=2048 Len=0 MSS=1460
15	14.639000	5.5.5.5	194.65.52.3	TCP	37836 > 80 [SYN] Seq=2796822245 Win=1024 Len=0 MSS=1460
16	14.649000	5.5.5.5	194.65.52.3	TCP	37836 > 554 [SYN] Seq=2796822245 Win=2048 Len=0 MSS=1460
17	14.656000	5.5.5.5	194.65.52.3	TCP	37836 > 22 [SYN] Seq=2796822245 Win=4096 Len=0 MSS=1460
18	14.664000	194.65.52.3	5.5.5.5	TCP	111 > 37836 [RST, ACK] Seq=0 Ack=2796822246 Win=0 Len=0
19	14.704000	5.5.5.5	194.65.52.3	TCP	37836 > 5900 [SYN] Seq=2796822245 Win=4096 Len=0 MSS=1460
20	14.717000	5.5.5.5	194.65.52.3	TCP	37836 > 256 [SYN] Seq=2796822245 Win=2048 Len=0 MSS=1460
21	14.736000	5.5.5.5	194.65.52.3	TCP	37836 > 443 [SYN] Seq=2796822245 Win=4096 Len=0 MSS=1460
22	14.755000	5.5.5.5	194.65.52.3	TCP	37836 > 135 [SYN] Seq=2796822245 Win=3072 Len=0 MSS=1460
23	14.756000	194.65.52.3	5.5.5.5	TCP	80 > 37836 [SYN, ACK] Seq=1928005857 Ack=2796822246 Win=4128
24	14.772000	5.5.5.5	194.65.52.3	TCP	37836 > 8888 [SYN] Seq=2796822245 Win=2048 Len=0 MSS=1460
25	14.773000	194.65.52.3	5.5.5.5	TCP	554 > 37836 [RST, ACK] Seq=0 Ack=2796822246 Win=0 Len=0
26	14.788000	5.5.5.5	194.65.52.3	TCP	37836 > 25 [SYN] Seq=2796822245 Win=3072 Len=0 MSS=1460
27	14.788000	194.65.52.3	5.5.5.5	TCP	22 > 37836 [RST, ACK] Seq=0 Ack=2796822246 Win=0 Len=0
28	14.807000	5.5.5.5	194.65.52.3	TCP	37836 > 53 [SYN] Seq=2796822245 Win=1024 Len=0 MSS=1460
29	14.813000	194.65.52.3	5.5.5.5	TCP	5900 > 37836 [RST, ACK] Seq=0 Ack=2796822246 Win=0 Len=0

Solução: verificação sistemática (Internet → DMZ)

```
C:\Documents and Settings\Cisco>nmap --system-dns 194.65.52.0/28
Starting Nmap 5.21 ( http://nmap.org ) at 2012-04-03 10:12 GMT Daylight Time
Nmap scan report for 194.65.52.1
Host is up (0.055s latency).
Not shown: 998 closed ports
PORT      STATE      SERVICE
23/tcp    open       telnet
80/tcp    filtered  http  ←
Nmap scan report for 194.65.52.2
Host is up (0.20s latency).
Not shown: 998 closed ports
PORT      STATE      SERVICE
23/tcp    open       telnet
80/tcp    filtered  http  ←
Nmap scan report for 194.65.52.3
Host is up (0.15s latency).
Not shown: 998 closed ports
PORT      STATE      SERVICE
23/tcp    open       telnet
80/tcp    open       http
Nmap scan report for 194.65.52.10
Host is up (0.30s latency).
All 1000 scanned ports on 194.65.52.10 are filtered
Nmap scan report for 194.65.52.14
Host is up (0.13s latency).
Not shown: 998 closed ports
PORT      STATE      SERVICE
23/tcp    open       telnet
80/tcp    filtered  http  ←
```

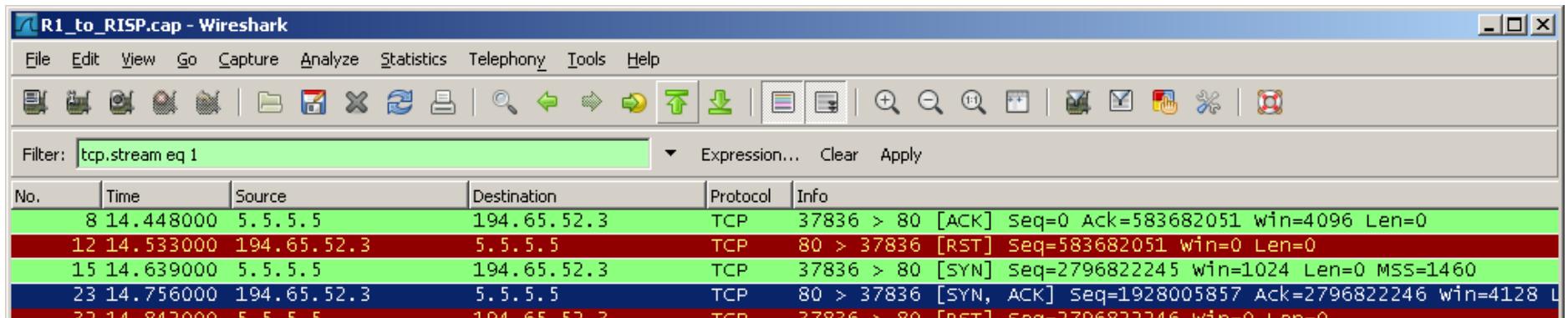
O nmap também
se engana!

Solução: verificação sistemática (Internet → DMZ)

No.	Time	Source	Destination	Protocol	Info
4	8.065000	5.5.5.5	194.65.52.1	ICMP	Echo (ping) request (id=0xe78f, seq(be/le)=0/0, ttl=50)
5	8.070000	5.5.5.5	194.65.52.2	ICMP	Echo (ping) request (id=0x6461, seq(be/le)=0/0, ttl=51)
6	8.084000	5.5.5.5	194.65.52.3	ICMP	Echo (ping) request (id=0x9c3e, seq(be/le)=0/0, ttl=40)
7	8.085000	5.5.5.5	194.65.52.4	ICMP	Echo (ping) request (id=0x7ea5, seq(be/le)=0/0, ttl=38)
8	8.092000	5.5.5.5	194.65.52.5	ICMP	Echo (ping) request (id=0xae51, seq(be/le)=0/0, ttl=42)
9	8.102000	5.5.5.5	194.65.52.6	ICMP	Echo (ping) request (id=0x590c, seq(be/le)=0/0, ttl=39)
10	8.109000	5.5.5.5	194.65.52.7	ICMP	Echo (ping) request (id=0xdfe5, seq(be/le)=0/0, ttl=46)
11	8.126000	5.5.5.5	194.65.52.8	ICMP	Echo (ping) request (id=0xb116, seq(be/le)=0/0, ttl=52)
12	8.196000	5.5.5.5	194.65.52.9	ICMP	Echo (ping) request (id=0xf153, seq(be/le)=0/0, ttl=54)
13	8.211000	5.5.5.5	194.65.52.10	ICMP	Echo (ping) request (id=0x42c2, seq(be/le)=0/0, ttl=52)
14	8.254000	194.65.52.1	5.5.5.5	ICMP	Echo (ping) reply (id=0xe78f, seq(be/le)=0/0, ttl=253)
15	8.367000	194.65.52.2	5.5.5.5	ICMP	Echo (ping) reply (id=0x6461, seq(be/le)=0/0, ttl=253)
16	8.378000	194.65.52.3	5.5.5.5	ICMP	Echo (ping) reply (id=0x9c3e, seq(be/le)=0/0, ttl=253)
17	8.385000	5.5.5.5	194.65.52.13	ICMP	Echo (ping) request (id=0x910f, seq(be/le)=0/0, ttl=47)
18	8.387000	5.5.5.5	194.65.52.14	ICMP	Echo (ping) request (id=0x5184, seq(be/le)=0/0, ttl=42)
19	8.389000	5.5.5.5	194.65.52.3	TCP	53168 > 443 [SYN] Seq=2410685217 Win=1024 Len=0 MSS=1460
20	8.393000	5.5.5.5	194.65.52.4	TCP	53168 > 443 [SYN] Seq=2410685217 Win=4096 Len=0 MSS=1460
21	8.404000	5.5.5.5	194.65.52.7	TCP	53168 > 443 [SYN] Seq=2410685217 Win=1024 Len=0 MSS=1460
22	8.425000	5.5.5.5	194.65.52.8	TCP	53168 > 443 [SYN] Seq=2410685217 Win=1024 Len=0 MSS=1460
23	8.425000	194.65.52.10	5.5.5.5	ICMP	Echo (ping) reply (id=0x42c2, seq(be/le)=0/0, ttl=62)
24	8.471000	5.5.5.5	194.65.52.9	TCP	53168 > 443 [SYN] Seq=2410685217 Win=2048 Len=0 MSS=1460
25	8.476000	194.65.52.14	5.5.5.5	ICMP	Echo (ping) reply (id=0x5184, seq(be/le)=0/0, ttl=254)
26	8.497000	5.5.5.5	194.65.52.12	ICMP	Echo (ping) request (id=0xead8, seq(be/le)=0/0, ttl=37)
27	8.505000	5.5.5.5	194.65.52.13	TCP	53168 > 443 [SYN] Seq=2410685217 Win=4096 Len=0 MSS=1460
28	8.512000	5.5.5.5	194.65.52.0	ICMP	Echo (ping) request (id=0xeb75, seq(be/le)=0/0, ttl=44)

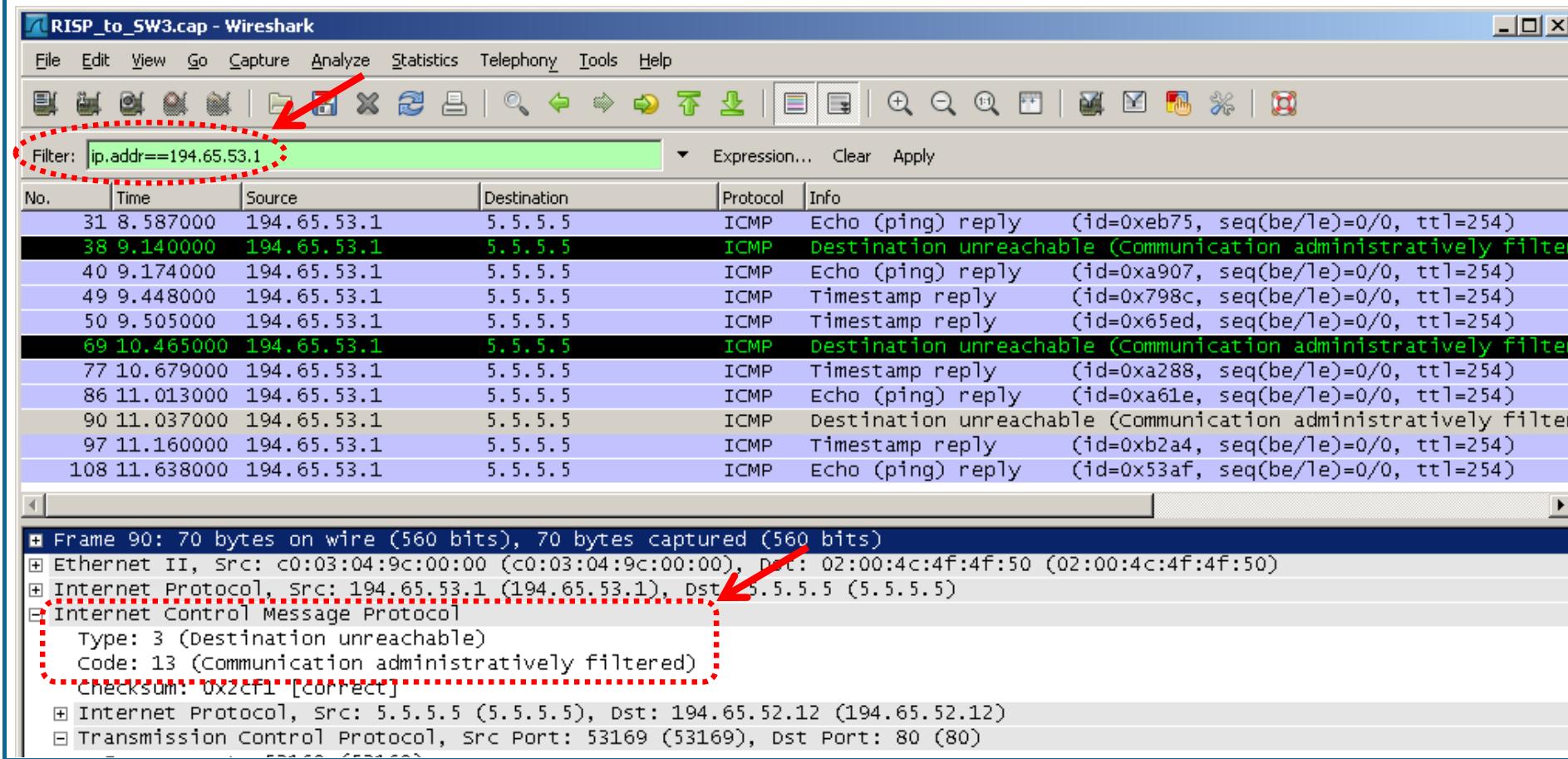
Solução: verificação sistemática

- Abordagem do *network mapper* (Nmap)
 1. Sonda utilização de endereço (ICMP Echo Request/Echo Reply)
 2. Tenta estabelecer sessões nos portos dos sistemas que respondem ao *ping* seguindo uma sequência pseudo-aleatória
 3. Para os serviços que respondem fecha de imediato a sessão aberta.



Solução: verificação sistemática (Internet → DMZ)

- Como são detectados os serviços *filtered*?

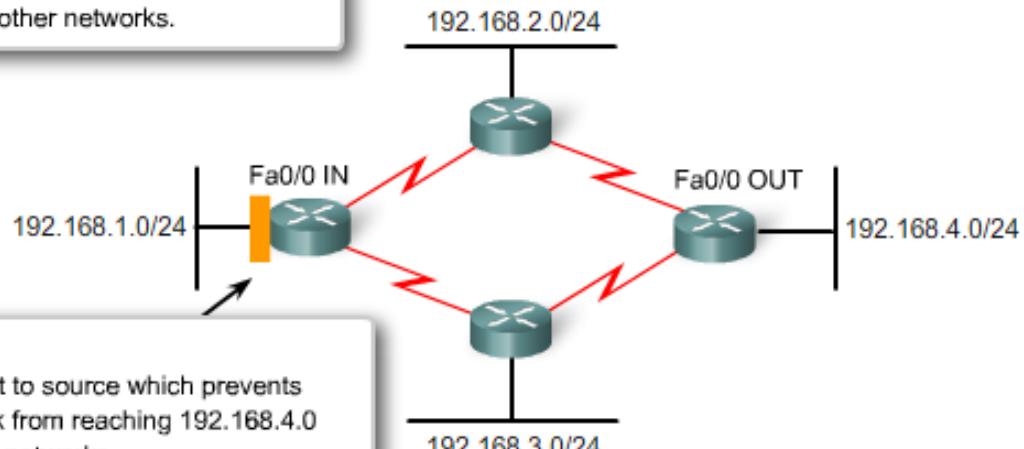


ACL Extended: localização

- Normalmente devem situar-se próximo da fonte do tráfego que controlam

Requirements:

Use Extended ACL to prevent traffic from the 192.168.1.0 network from entering the 192.168.4.0 network but allow it to reach other networks.



ACL

```
access-list 109 deny ip 192.168.1.0 0.0.0.255 192.168.4.0 0.0.0.255  
access-list 109 permit ip any any
```

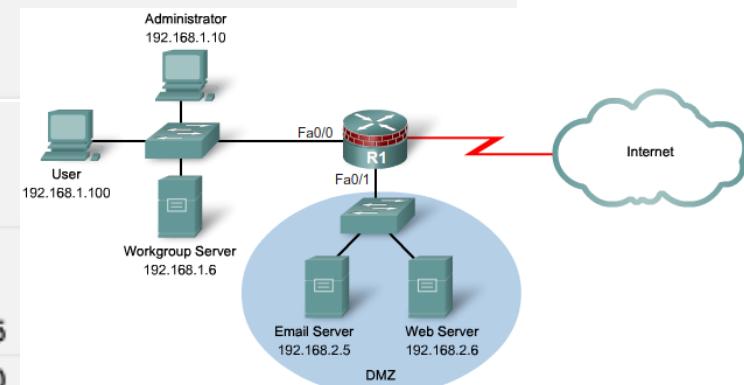


ACL Extended: identificação por nome (named)

- Identificação com base em nomes (named ACL)
 - Exemplo

```
R1(config)# ip access-list extended ACL-1
R1(config-ext-nacl)# remark LAN ACL
R1(config-ext-nacl)# deny ip host 192.168.1.6 any
R1(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 any established
R1(config-ext-nacl)# deny ip any any
R1(config-ext-nacl)# exit

R1(config)# interface Fa0/0
R1(config-if)# ip access-group ACL-1 in
R1(config-if)# exit
R1(config)# ip access-list extended ACL-2
R1(config-ext-nacl)# remark DMZ ACL
R1(config-ext-nacl)# permit tcp any host 192.168.2.5 eq 25
R1(config-ext-nacl)# permit tcp any host 192.168.2.6 eq 80
R1(config-ext-nacl)# deny ip any any
R1(config-ext-nacl)# exit
R1(config)# interface Fa0/1
R1(config-if)# ip access-group ACL-2 out
R1(config-if)# exit
```



ACL Extended: opção established

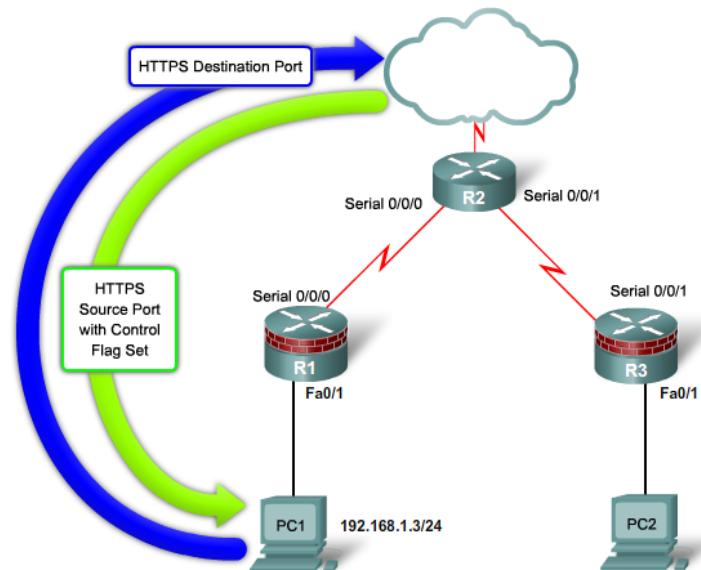
- “established” é uma opção das ACL que controlam explicitamente tráfego TCP
 - Opção introduzida em 1995
 - Não converte o *router* numa *firewall statefull*
 - IOS continua a não guardar informação acerca do estado das sessões
- Que tráfego verifica a condição “established”?
 - Segmentos TCP com a *flag* ACK ou RST activas
 - i.e., todos os segmentos excepto o segmento inicial (SYN)

TCP Segment												
Bit offset	Bits 0-3	4-7	8-15	16-31								
0	Source Port								Destination Port			
32	Sequence Number											
64	Acknowledgement Number											
96	Checksum	Reserved	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size	
128	Checksum		Urgent Pointer									
160	Options (optional)											
160/192+	Data											



ACL Extended: exemplo da opção established

- No *router/firewall* R1 do diagrama lateral permitir que entre tráfego de sessões HTTPS (porto TCP 443) iniciadas pelos clientes 192.168.1.0/24 bem como de sessões SSH (TCP 22) destinadas exclusivamente ao PC1.

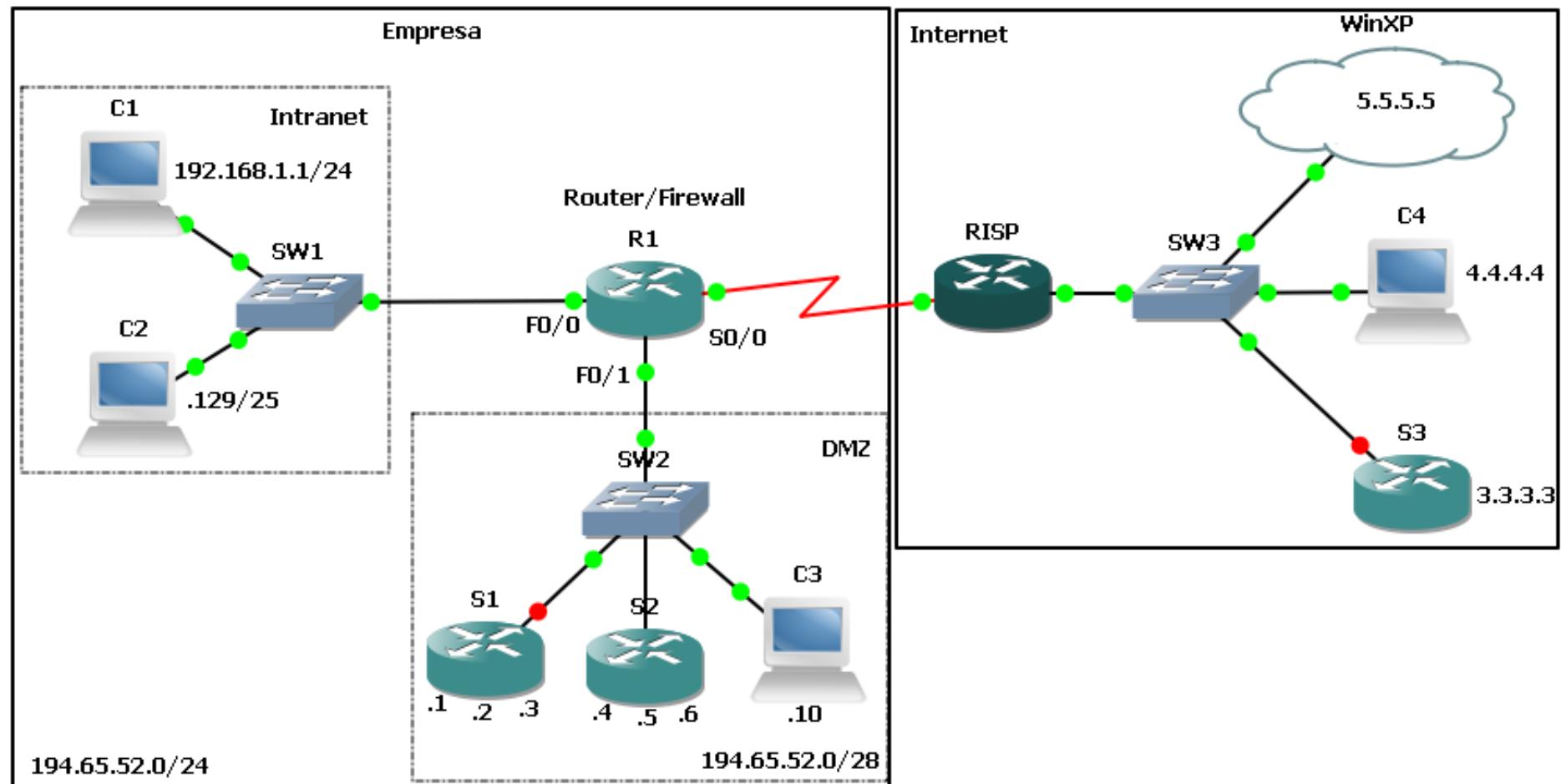


```
R1(config)# access-list 100 permit tcp any eq 443 192.168.1.0 0.0.0.255
established
R1(config)# access-list 100 permit tcp any 192.168.1.3 0.0.0.0 eq 22
R1(config)# access-list 100 deny ip any any
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 100 in
```

- O que sucederia se a ACL não possuísse “established”?



Exercício



Exercício

- Políticas de segurança:
 - a) Evitar *spoofing* tanto para proteger a empresa como para evitar que a própria empresa seja fonte deste tipo de ataques.
 - b) Da Internet apenas deve ser possível estabelecer sessões TCP com a DMZ nos seguintes serviços: {DNS, porto 53, 194.65.52.1; SMTP, porto 25, 194.65.52.2; HTTP, porto 80, 194.65.52.3/4; POP3S, porto 995, 194.65.52.5}.
 - c) De toda a empresa deve ser possível estabelecer qualquer tipo de sessão TCP com a Internet. Duas restrições devem ser verificadas:
 - i. os acessos HTTP ao exterior devem provir exclusivamente do proxy (194.65.52.6);
 - ii. apenas o servidor de e-mail (194.65.52.2) pode estabelecer sessões SMTP com o exterior;

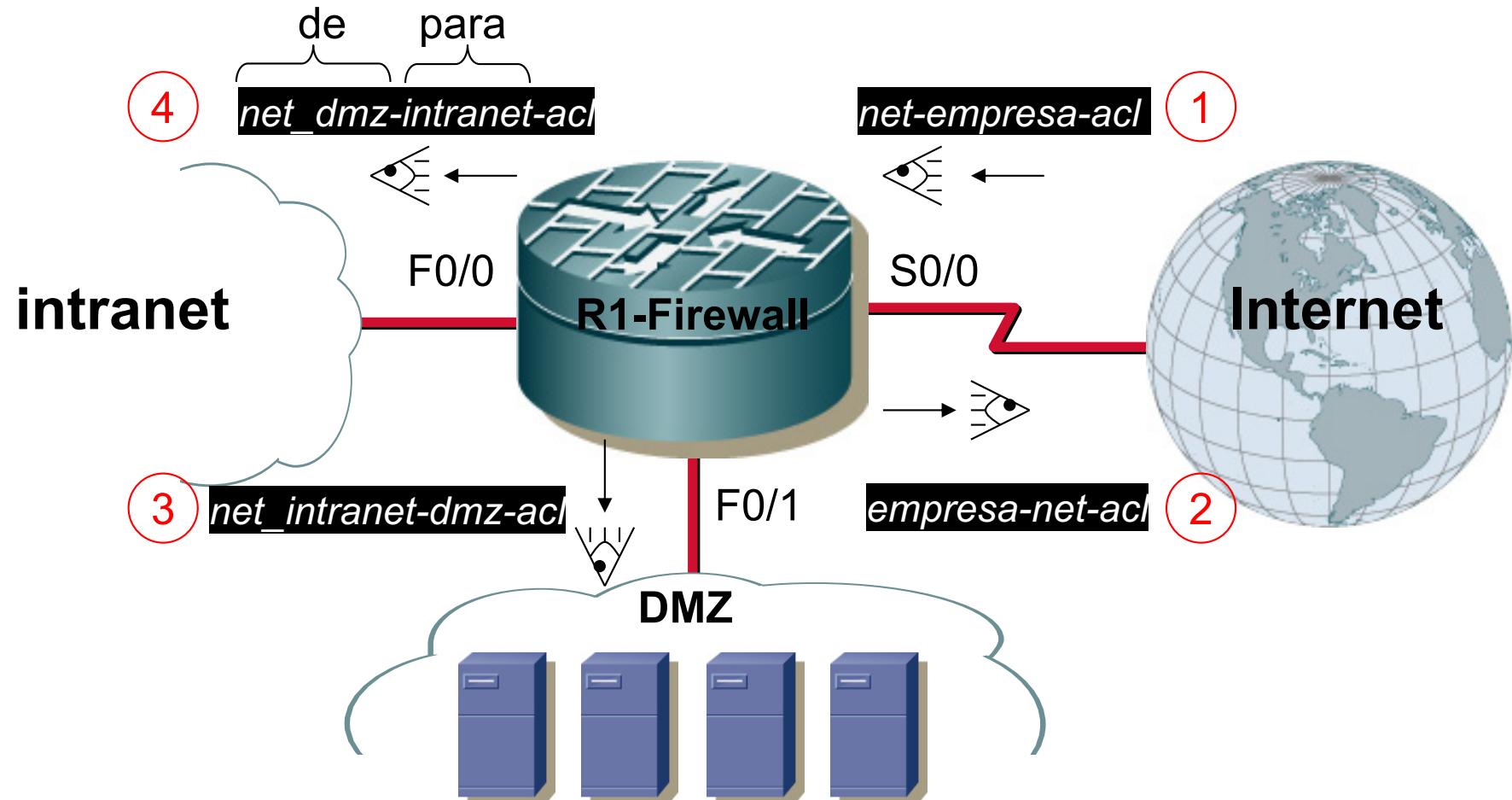


Exercício

- Políticas de segurança (cont.):
 - iii. a troca de tráfego UDP com o exterior encontra-se restringida ao serviço DNS (porto 53, 194.65.52.1) da DMZ.
 - d) Da intranet deve ser possível estabelecer qualquer sessão TCP com a DMZ. Deve ainda ser permitido trocar tráfego UDP destinado ao serviço DNS.
 - e) Da DMZ não deve ser permitido estabelecer sessões TCP para a intranet, devendo no entanto ser possível fazê-lo em sentido inverso.
 - f) O tráfego ICMP relevante para o bom funcionamento e diagnóstico da rede deve poder circular livremente.
 - g) O acesso ao *router* fronteiro deve estar limitado a sessões SSH e HTTPS estabelecidas a partir do interior da empresa.



Solução



Solução

1. ACL “net-empresa-acl”

- Tráfego liminarmente impedido de entrar na empresa

```
ip access-list extended net-empresa-acl
  remark Tráfego liminarmente impedido de entrar na empresa.
```

!a) Tráfego que constitua uma ameaça de spoofing

```
deny ip 194.65.52.0 0.0.0.255 any
deny ip 224.0.0.0 15.255.255.255 any
deny ip 127.0.0.0 0.255.255.255 any
deny ip host 0.0.0.0 any
deny ip 10.0.0.0 0.255.255.255 any
deny ip 172.16.0.0 0.15.255.255 any
deny ip 192.168.0.0 0.0.255.255 any
permit ip any any
exit
```

```
interface s1/0
  ip access-group net-empresa-acl in
```



Solução

2. ACL “empresa-net-acl ”

- Tráfego autorizado a sair da empresa

```
ip access-list extended empresa-net-acl
  remark Tráfego autorizado a sair da empresa
  !f) Tráfego ICMP relevante ao bom funcionamento e diagnóstico da rede
  permit icmp 194.65.52.0 0.0.0.255 any echo
  permit icmp 194.65.52.0 0.0.0.255 any echo-reply
  permit icmp 194.65.52.0 0.0.0.255 any unreachable
  permit icmp 194.65.52.0 0.0.0.255 any ttl-exceeded

  !c-i) Sessões HTTP exclusivamente estabelecidas pelo proxy
  permit tcp host 194.65.52.6 any eq www
  deny tcp any any eq www

  !c-ii) Sessões SMTP exclusivamente estabelecidas pelo servidor SMTP
  permit tcp host 194.65.52.2 any eq smtp
  deny tcp any any eq smtp
```



Solução

2. ACL “empresa-net-acl” (cont.)

- Tráfego autorizado a sair da empresa

```
!c-iii) Sessões DNS exclusivamente estabelecidas pelo servidor DNS  
permit tcp host 194.65.52.1 any eq domain  
deny tcp any any eq domain  
permit udp host 194.65.52.1 any eq domain  
deny udp any any eq domain  
  
!Respostas UDP do servidor DNS  
permit udp host 194.65.52.1 eq domain any  
  
!a,c) O restante tráfego TCP que não constitua spoofing  
permit tcp 194.65.52.0 0.0.0.255 any  
deny ip any any  
exit  
  
interface s1/0  
ip access-group empresa-net-acl out
```



Solução

3. ACL “net_intranet-dmz-acl” (cont.)

- Tráfego autorizado a entrar na DMZ

```
ip access-list extended net_intranet-dmz-acl
  remark Tráfego autorizado a entrar na DMZ
```

!f) Tráfego ICMP relevante ao bom funcionamento e diagnóstico da rede

```
  permit icmp any any echo
  permit icmp any any echo-reply
  permit icmp any any unreachable
  permit icmp any any ttl-exceeded
```

!c) Tráfego de retorno de sessões TCP iniciadas na DMZ

!Nota: sessões da DMZ destinadas à intranet são vedadas em

```
!net_dmz-intranet-acl
```

```
  permit tcp any any established
```

!d) Acesso interno a serviços (DMZ) da empresa

```
  permit tcp 194.65.52.0 0.0.0.255 any
  permit udp 194.65.52.0 0.0.0.255 host 194.65.52.1 eq domain
```



Solução

3. ACL “net_intranet-dmz-acl” (cont.)

- Tráfego autorizado a entrar na DMZ

```
!b) Acesso público a serviços (DMZ) da empresa
permit tcp any host 194.65.52.1 eq domain
permit udp any host 194.65.52.1 eq domain
permit tcp any host 194.65.52.2 eq smtp
permit tcp any host 194.65.52.3 eq www
permit tcp any host 194.65.52.4 eq www
permit tcp any host 194.65.52.5 eq 995
deny ip any any log
exit

interface f0/1
  ip access-group net_intranet-dmz-acl out
end
```



Solução

4. ACL “net_dmz-intranet-acl”

- Tráfego autorizado a entrar na intranet

```
ip access-list extended net_dmz-intranet-acl
  remark Que tráfego pode entrar na intranet?
```

!f) Tráfego ICMP relevante ao bom funcionamento e diagnóstico da rede

```
  permit icmp any any echo
  permit icmp any any echo-reply
  permit icmp any any unreachable
  permit icmp any any ttl-exceeded
```

!e) Tráfego de retorno de sessões TCP iniciadas na intranet

```
  permit tcp any any established
```

!d) Respostas do servidor DNS da DMZ

```
  permit udp host 194.65.52.1 eq domain any
  deny ip any any log
  exit
```



Solução

4. ACL “net_dmz-intranet-acl” (cont.)

- Tráfego autorizado a entrar na intranet

```
interface f0/0
  ip access-group net_dmz-intranet-acl out
end
```



Solução

- Proteger devidamente o acesso ao *router* fronteiro

```
ip access-list standard router_access-acl
    remark Sistemas autorizados a gerir este router
    permit 194.65.52.0 0.0.0.255
    permit 10.0.0.0 0.255.255.255
    permit 172.16.0.0 0.15.255.255
    permit 192.168.0.0 0.0.255.255
    deny any
exit

line vty 0 4
    transport input ssh
    access-class router_access-acl in

ip http access-class router_access-acl in
```



Reflexive ACLs

DEIS

Fragilidades da primeira geração de *firewalls* IOS

- Com uma ferramenta como Nmap ou o Collasoft PacketBuilder é possível alterar o valor das *flags* TCP e, deste modo, fazer passar através de uma ACL *established* tráfego ilícito
 - Apesar de ser possível explorar estes “buracos” (*hole*) na *firewall* uma solução baseada na opção *established* deve ser considerada mais segura que a tradicional abertura nos dois sentidos.
- Adicionalmente a opção *established* apenas é aplicável a tráfego TCP. Tráfego UDP, ICMP ou outro não é abrangido!



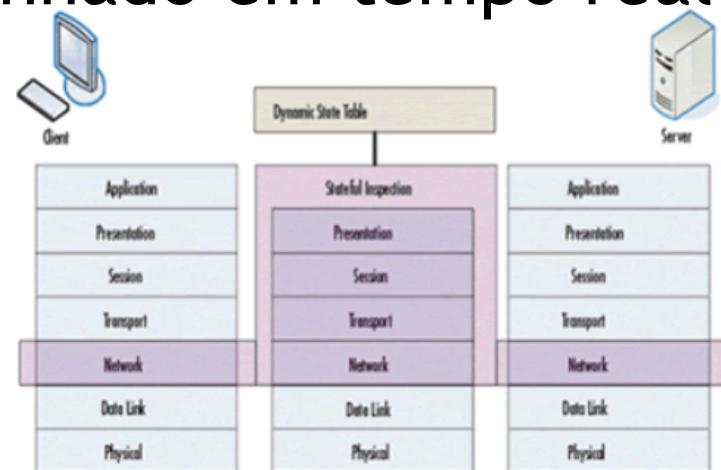
Segunda geração de firewalls IOS (ACLs reflexivas)

- 1996: Um ano após a introdução da opção *established* a Cisco introduz as ACL reflexivas.
- Características das ACL reflexivas:
 - Conceito de sessão assenta em mais factores
 - Endereço fonte (SA), Endereço destino (DA)
 - Porto fonte (SP), Porto destino (DP)
 - Flags TCP e outros campos
 - Introdução de filtros temporários removidos após o encerramento de cada sessão ou de um *timeout*
 - As Access Control Entries (ACEs) criadas dinamicamente são inseridas nas ACL extended das interfaces de saída



Segunda geração de firewalls IOS (ACLs reflexivas)

- A grande limitação das ACL *standard* e *extended* é que não foram desenhadas para manter o estado das sessões que permitiam estabelecer
- As ACL reflexivas vieram representar uma solução mais segura face à opção *established* uma vez que passa a ser mantido e acompanhado em tempo real o estado de cada sessão que é permitida por determinada ACL
 - Com as ACL reflexivas a Cisco abriu a sua era de *Stateful firewalls*

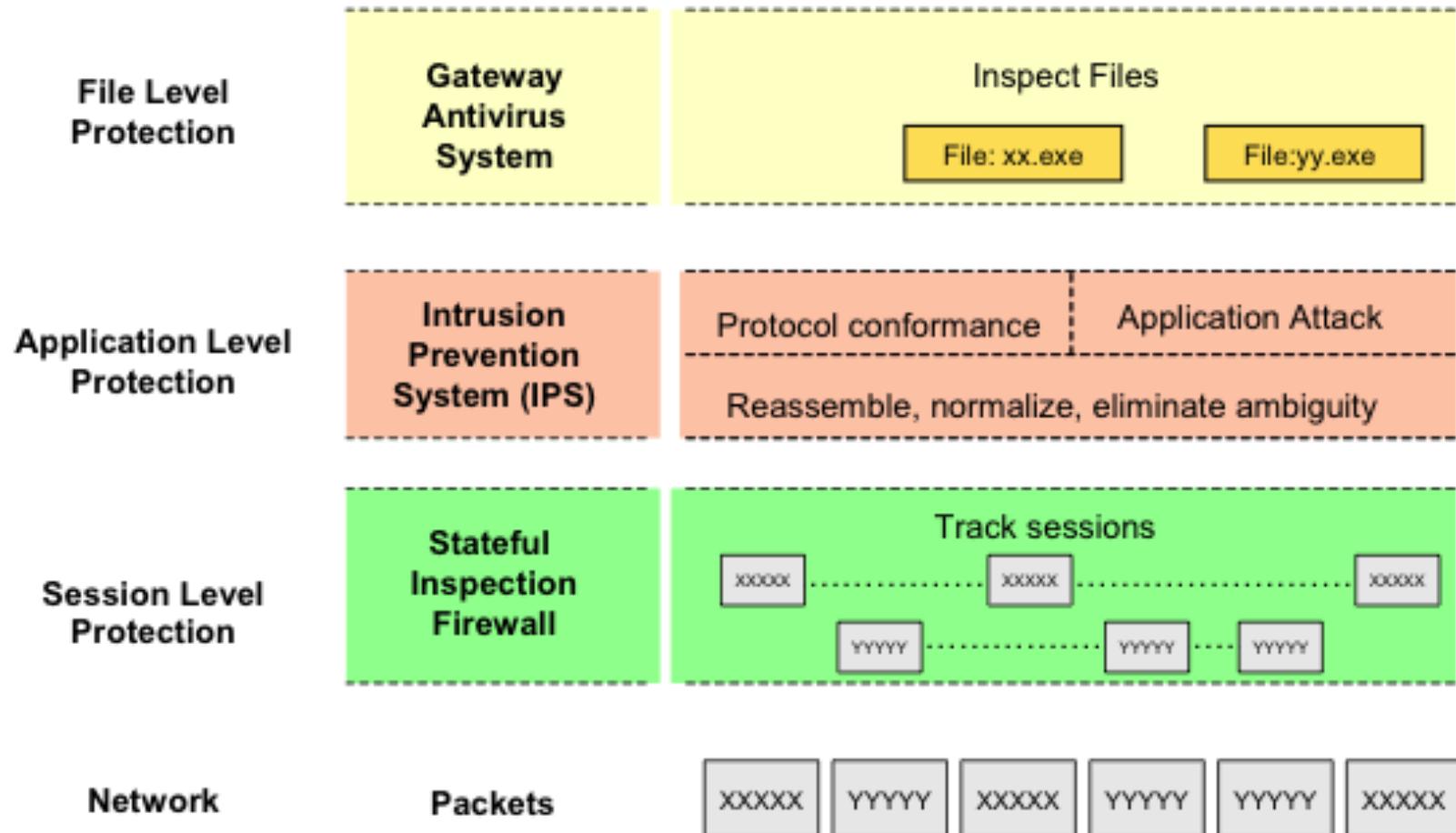


Mecanismos de segurança

	Packet Level Protection	Session Level Protection	Application Level Protection	File Level Protection
Examples	Packet filtering (router ACLs or stateless firewalls)	Stateful inspection firewalls	Intrusion prevention systems (IPS) and proxy firewalls	Gateway antivirus
Mechanism	Examine packet header	Examine packet header and control fields	Examine application fields	Examine files inside application traffic
Protocol and Application Coverage	N.A. packet level	Large	Medium	Small (email, web and file transfers)
Protection Provided	Client-to-server and server-to-client	Client-to-server and server-to-client	Mainly client-to-server	Mainly server-to-client
Relative Performance	High	High	Medium	Low



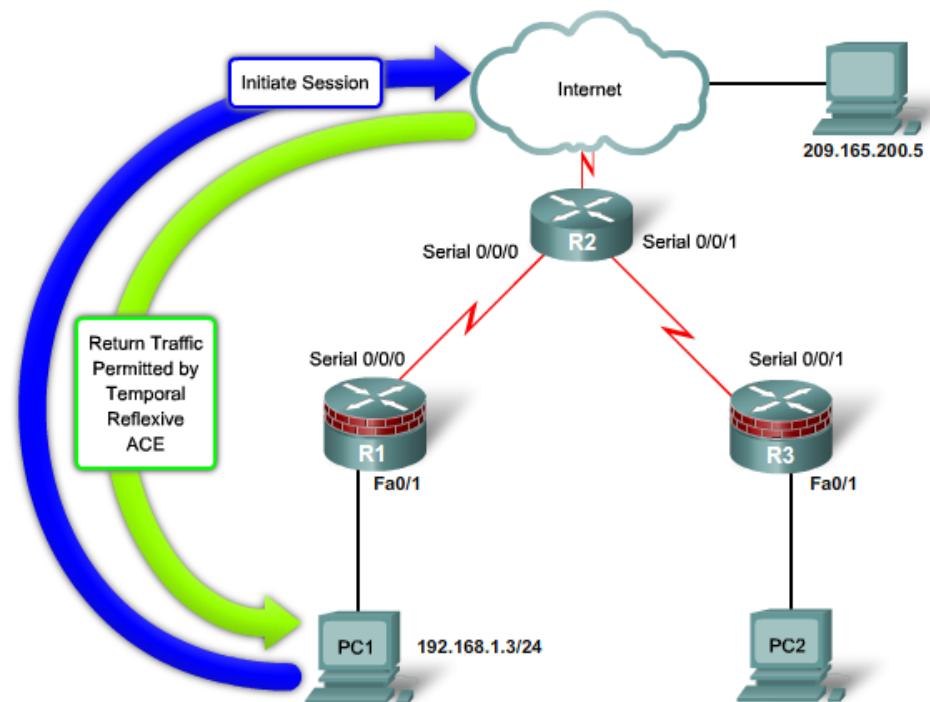
Mecanismos de segurança



ACLs reflexivas: funcionamento

- Introduzidas em 1996
- *Extended named ACLs*
 - keyword *reflect* em decisões *permit*
- Quando as NACL reflexivas são cruzadas por tráfego aceite o IOS cria as ACEs necessárias para aceitar o tráfego de retorno nos pontos *evaluate* especificados.

```
permit host 209.165.200.5 eq 23 host 192.168.1.3 eq 11000
```



Em resposta à abertura de uma sessão a partir do PC1 para o destino 209.165.200.5 é criada dinamicamente em R1 uma ACL



ACLs reflexivas: configuração

1. Criar e aplicar uma ACL que inspecione a saída de sessões de interesse e gere as ACEs temporárias necessárias para permitir a entrada do reflexo (i.e., de tráfego de retorno)

```
Router(config)# ip access-list extended internal_ACL_name
Router(config-ext-nacl)# permit protocol source-addr [source-mask]
[operator operand] destination-addr [destination-mask] [operator
operand] [established] reflect reflexive_ACE_name [timeout seconds]
```

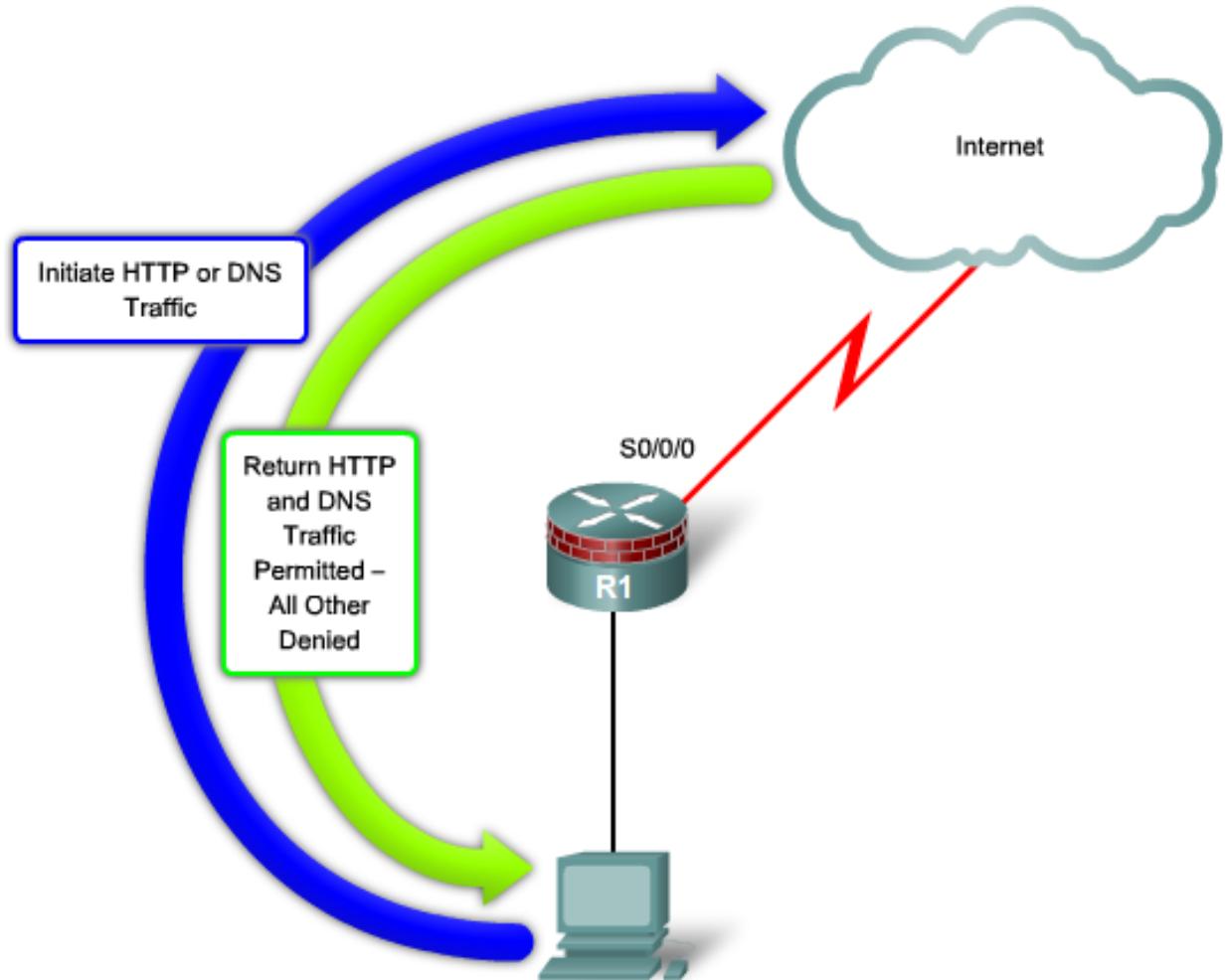
2. Criar e aplicar uma ACL reflexiva que inspecione a entrada do tráfego de retorno

```
Router(config)# ip access-list extended external_ACL_name
Router(config-ext-nacl)# evaluate reflexive_ACE_name
```



ACLs reflexivas: exercício

- Programe o *router* de fronteira R1 de modo que permita aos utilizadores internos usar normalmente o seu browser para “surfar” na Internet.



ACLs reflexivas: solução

1. Permitir em R1 que terminais da rede interna iniciem sessões HTTP e consultas DNS externas

```
R1(config)# ip access-list extended internal_ACL
R1(config-ext-nacl)# permit tcp any any eq 80 reflect web-reflex-ACE
R1(config-ext-nacl)# permit udp any any eq 53 reflect dns-reflex-ACE timeout
10
R1(config)# interface s0/0/0
R1(config-if)# description connection to the ISP.
R1(config-if)# ip access-group internal_ACL out
```

2. Permitir em R1 que o tráfego de retorno das sessões mencionadas entre na rede interna

```
R1(config)# ip access-list extended external_ACL
R1(config-ext-nacl)# evaluate web-reflex-ACE
R1(config-ext-nacl)# evaluate dns-reflex-ACE
R1(config-ext-nacl)# deny ip any any
R1(config)# interface s0/0/0
R1(config-if)# ip access-group external_ACL in
```

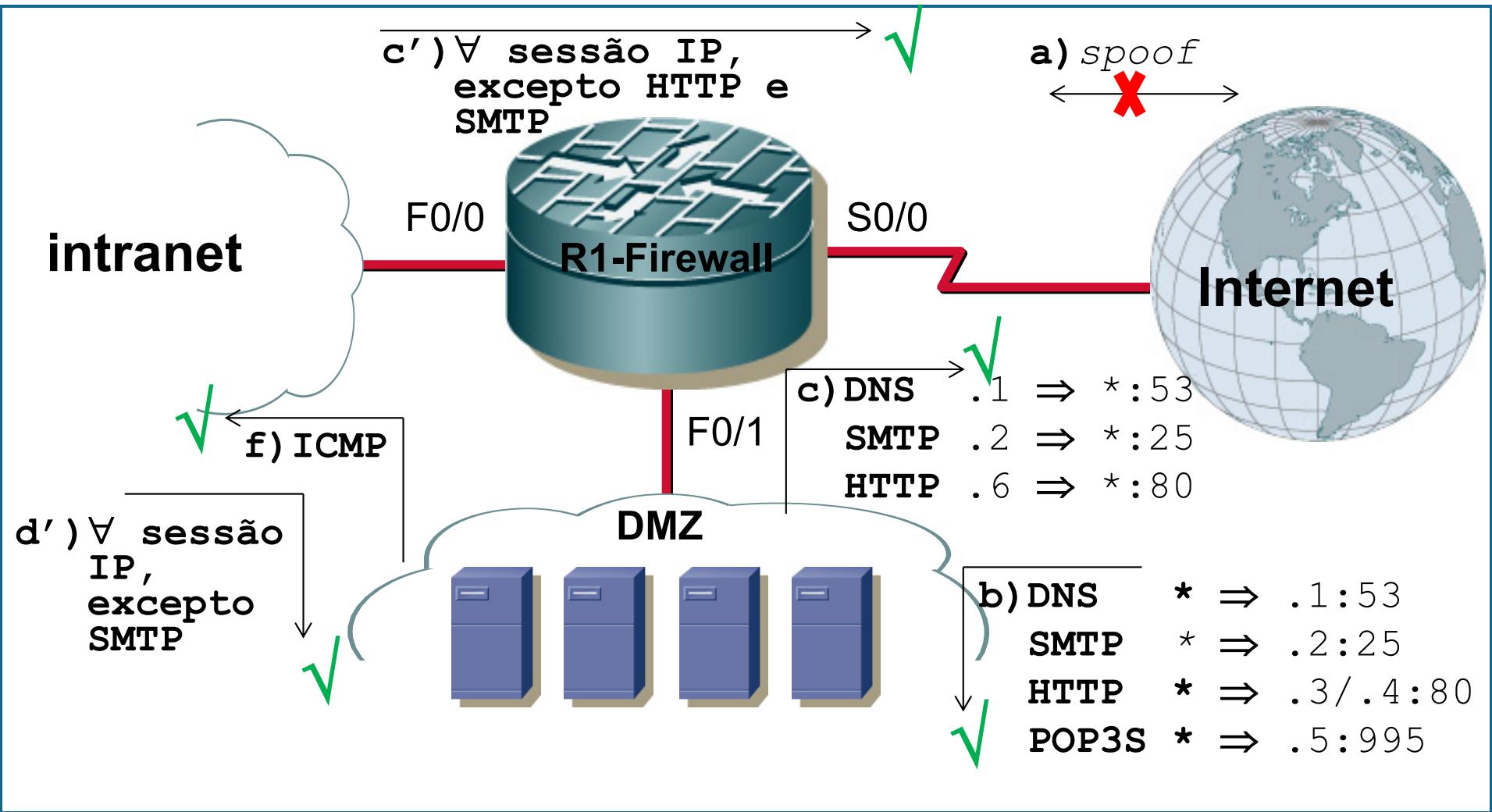


ACLs reflexivas: desafio

- Implemente as políticas de segurança descritas no módulo anterior com ACLs reflexivas atendendo à seguinte actualização de requisitos:
 - c) A abertura de sessões TCP ao exterior deve ficar limitada aos terminais da intranet (deste modo reduz-se o perigo de um ataque bem sucedido a um qualquer sistema da DMZ). No entanto, além de sessões TCP, deve passar a ser possível estabelecer sessões de outro qualquer protocolo assente em IP (UDP, ICMP, ...).
 - Nota: Restrições a manter: i) HTTP, ii) SMTP e iii) DNS.
 - d) Além de sessões TCP, deve passar a ser possível estabelecer sessões de outro qualquer protocolo assente em IP (UDP, ICMP, ...) excepto SMTP.



ACLs reflexivas: desafio



ACLs reflexivas: desafio

- Solução: TPC



Dynamic ACLs (≡ Lock-and-Key)

DEIS

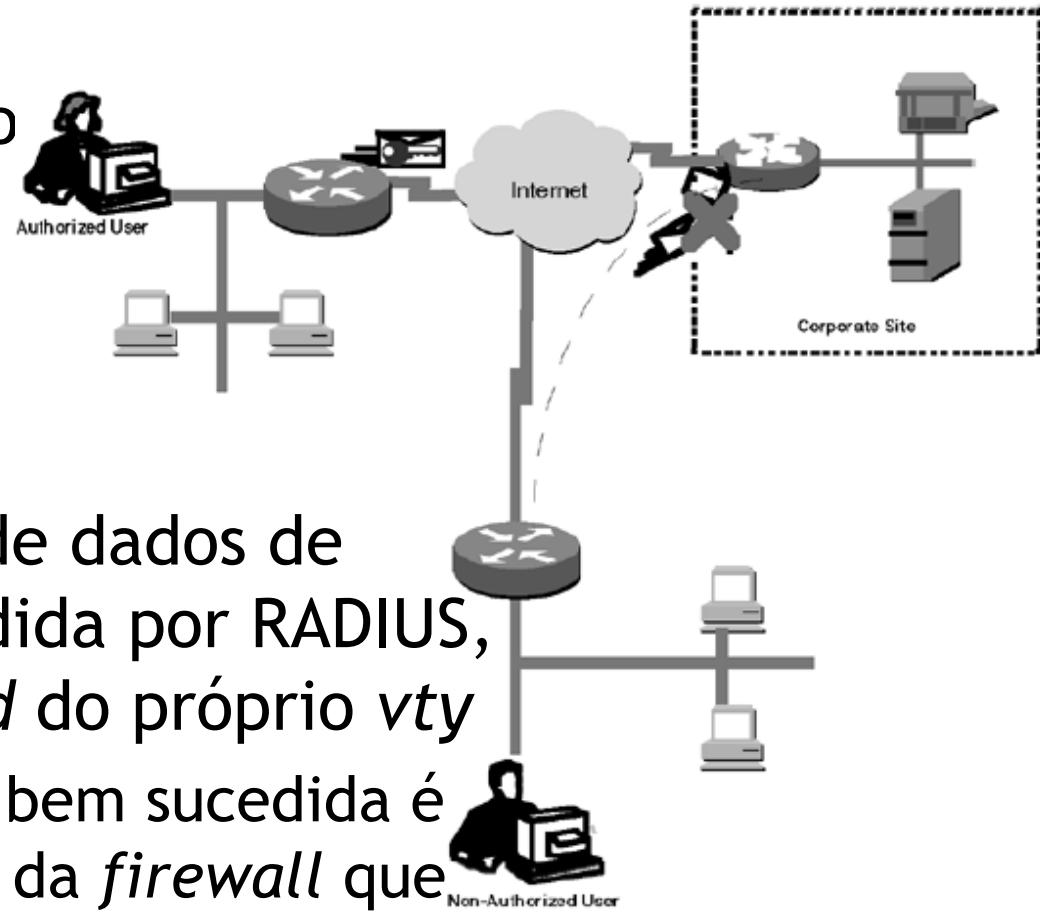
ACLs dinâmicas

- Também conhecidas por *Lock-and-key ACLs*
- 1996(Feb): IOS \geq 11.1
 - Surgiu antes das ACLs reflexivas e da capacidade de *logging* das ACL terem sido introduzidas
- O objectivo deste tipo de ACL foi fornecer uma ferramenta capaz de, de forma dinâmica, abrir temporariamente “excepções” na *firewall* em resposta a pedidos devidamente autenticados
 - ++: autenticação individual e independente da aplicação
 - --: a) potencialmente vulnerável ao *spoofing* (evitável com encriptação) ; b) políticas individuais não são suportadas



ACLs dinâmicas: funcionamento

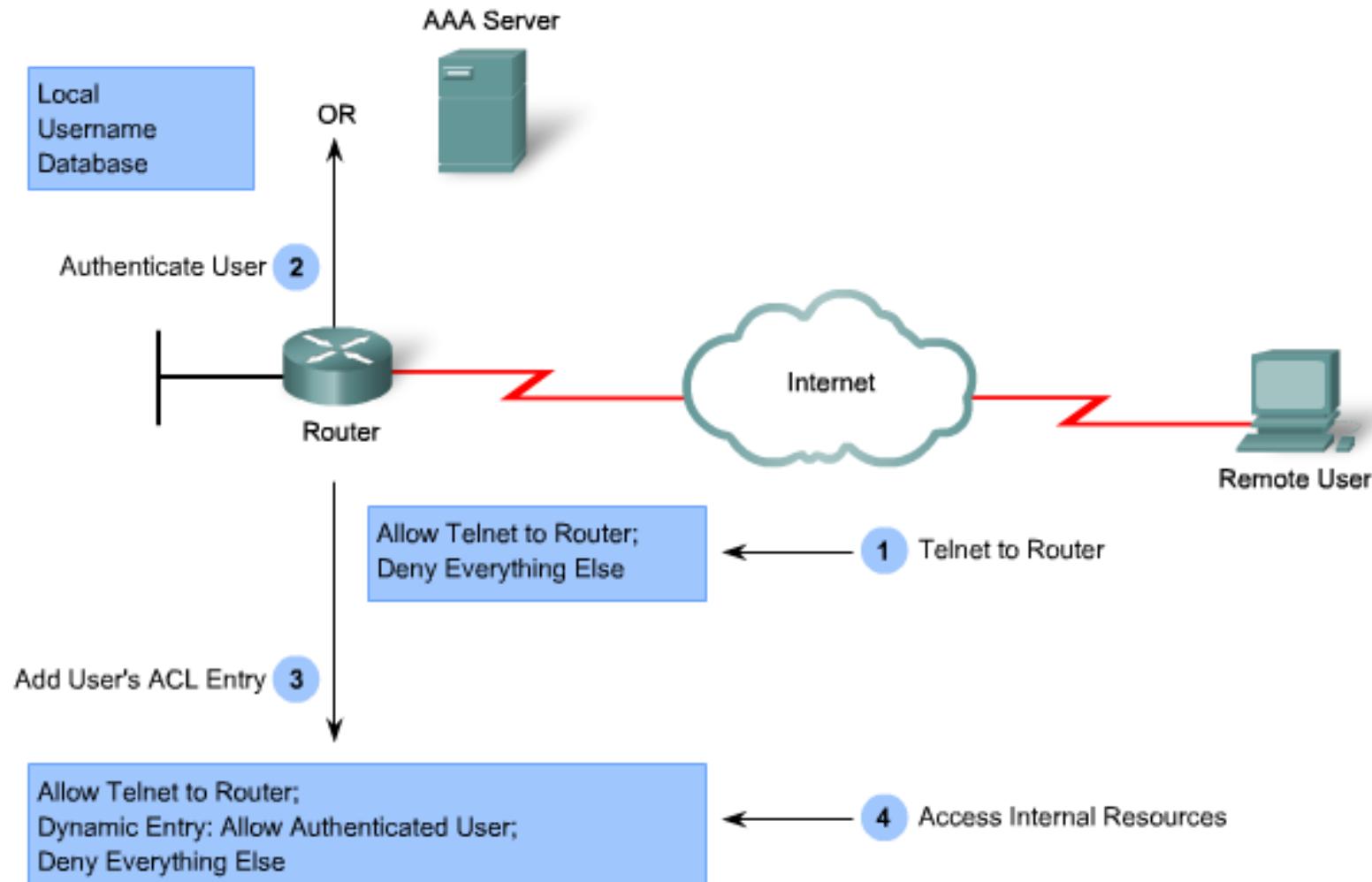
1. O utilizador deve estabelecer uma sessão SSH/telnet para o *router* e autenticar-se com o par *user/password*



2. O IOS consulta a base de dados de utilizadores local, acedida por RADIUS, TACACS+ ou a *password* do próprio vty
 - Sendo a autenticação bem sucedida é aberta uma passagem da *firewall* que permite aceder a determinado serviço.



ACLs dinâmicas: funcionamento



ACLs dinâmicas: configuração

1. Criar uma Extended ACLs

- São suportadas ACLs identificadas por nº e nome
- Uma das primeiras entradas deve permitir explicitamente sessões *telnet* e/ou SSH para o IP da interface do *router* por onde entra o pedido
- Uma das seguintes entradas deve indicar o *placeholder* onde a ACL dinâmica será considerada depois da autenticação ser bem sucedida.

2. Definir o esquema de autenticação usado

- *Username database*, RADIUS, TACACS+,

3. Activar o método de autenticação dinâmico



ACLs dinâmicas: exemplo

- Definição do *template* na ACL dinâmica

```
Router(config)# access-list {100-199} dynamic dynamic_ACL_name [timeout  
minutes] {permit | deny} protocol source-addr [source-wildcard] [operator  
operand] destination-addr [destination-wildcard] [operator operand]  
[established]
```

dynamic dynamic-name

Identifies this access list as a dynamic access list. Refer to lock-and-key access documented in the “Configuring Lock-and-Key Security (Dynamic Access Lists)” chapter in the *Cisco IOS Security Configuration Guide*.

timeout minutes

Specifies the absolute length of time, in minutes, that a temporary access list entry can remain in a dynamic access list. The default is an infinite length of time and allows an entry to remain permanently. Refer to lock-and-key access documented in the “Configuring Lock-and-Key Security (Dynamic Access Lists)” chapter in the *Cisco IOS Security Configuration Guide*.



ACLs dinâmicas: exemplo

- Activar a criação de entradas temporárias na ACL dinâmica

```
access-enable [host] [timeout minutes]
```

host (Optional) Tells the software to enable access only for the host from which the Telnet session originated. If not specified, the software allows all hosts on the defined network to gain access. The dynamic access list contains the network mask to use for enabling the new network.

timeout minutes (Optional) Specifies an idle timeout for the temporary access list entry. If the access list entry is not accessed within this period, it is automatically deleted and requires the user to authenticate again. The default is for the entries to remain permanently. We recommend that this value equal the idle timeout set for the WAN connection.

- Nota: Este é o comando a programar para execução automática assim que o *user* se autenticar.



ACLs dinâmicas: exemplo

- Activar a criação de entradas temporárias na ACL dinâmica assim que o utilizador se autentique

```
line vty i j  
autocommand access-enable [host] [timeout minutes]
```



ACLs dinâmicas: configuração

- Eliminação de entradas temporárias em ACLS dinâmicas

```
clear access-template {access-list-number | name} template-name {source-address source-wildcard-bit | any | host {hostname | source-address}} {destination-address dest-wildcard-bit | any | host {hostname | destination-address}} [timeout minutes]
```

access-list-number Number of the dynamic access list. The ranges are from 100 to 199 and from 2000 to 2699.

name Name of an IP access list.

- The name cannot contain a space or quotation mark, and must begin with an alphabetic character to avoid ambiguity with numbered access lists.

template-name Name of a dynamic access list.

source-address Source address in a dynamic access list.

- All other attributes are inherited from the original access-list entry.

source-wildcard-bit Source wildcard bits.

any Specifies any source hostname.



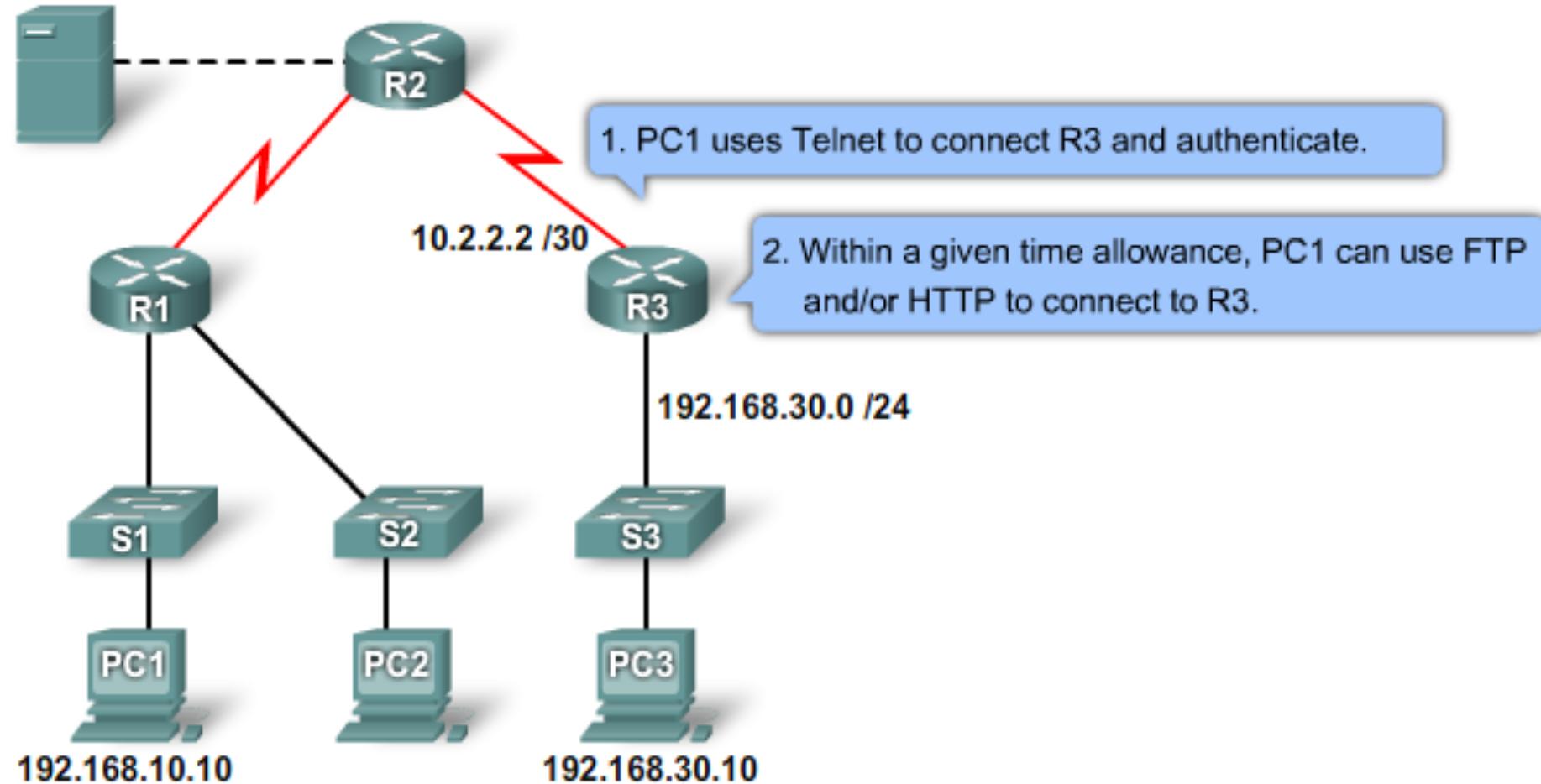
ACLs dinâmicas: configuração

- Eliminação de entradas temporárias em ACLS dinâmicas

host	Specifies a specific source host.
<i>hostname</i>	Name of the host.
<i>destination-address</i>	Destination address in a dynamic access list. <ul style="list-style-type: none">• All other attributes are inherited from the original access-list entry.
<i>dest-wildcard-bit</i>	Destination wildcard bits.
timeout minutes	(Optional) Specifies a maximum time limit, in minutes, for each entry within this dynamic list. The range is from 1 to 9999. <ul style="list-style-type: none">• This is an absolute time, from creation, that an entry can reside in the list. The default is an infinite time limit and allows an entry to remain permanently.



ACLs dinâmicas: exemplo



ACLs dinâmicas: exemplo

- Solução

Step 1

```
R3(config)# username Student password 0 cisco
```

Step 2

```
R3(config)# access-list 101 permit tcp any host 10.2.2.2  
eq telnet  
R3(config)# access-list 101 dynamic testlist timeout 15  
permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
```

Step 3

```
R3(config)# interface serial 0/0/1  
R3(config-if)# ip access-group 101 in
```

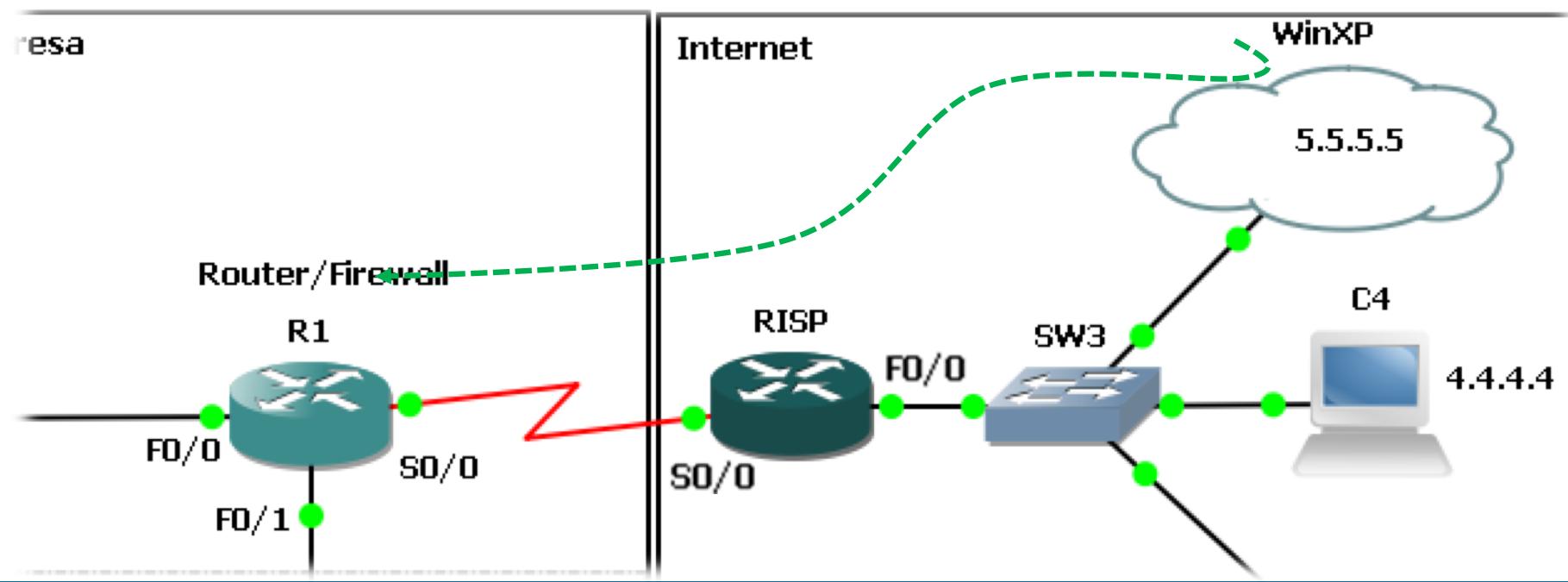
Step 4

```
R3(config)# line vty 0 4  
R3(config-line)# login local  
R3(config-line)# autocommand access-enable host timeout 5
```



ACLs dinâmicas: exercício

- Programe RISP de modo que apenas terminais da rede 5.5.5.0/24 possam contactar o “servidor” R1 quando usados por utilizadores registados e autenticados em RISP



ACLs dinâmicas: exercício

1. Registar um (ou mais) utilizadores em R-ISP

```
R-ISP(config)#username user1 secret user1
```

- Nota: pode usar-se qualquer uma das possibilidades do AAA

2. Vedar o tráfego na interface de entrada F0/0, abrindo exceção à autenticação e a entradas dinâmicas que decorram da mesma

```
R-ISP(config)#access-list 101 permit tcp any host 5.5.5.254 eq telnet
R-ISP(config)#access-list 101 dynamic user1-list timeout 15 permit ip
5.5.5.0 0.0.0.255 host 194.65.52.254
R-ISP(config)# interface f0/0
R-ISP(config-if)#ip access-group 101 in
```



Absolute timeout: Depois de autenticado o utilizador apenas dispõe de 15 minutos para dar início a uma eventual sessão. Ao fim dos 15 minutos o acesso é vedado mesmo que a sessão esteja em curso

ACLs dinâmicas: exercício

3. Programar o serviço *telnet* para autenticação local e execução automática do comando que despoletará a criação de entrada temporária na ACL dinâmica da interface por onde entrou a sessão *telnet*

```
R-ISP(config-if)#line vty 0 4
R-ISP(config-line)#login local
R-ISP(config-line)#autocommand access-enable host timeout 5
```

A posição onde é colocado o *template* é importante pois será aí que são criadas as entradas dinâmicas.

4. Inspecionar a ACL dinâmica criada

```
R-ISP#show access-lists
Extended IP access list 101
    10 permit tcp any host 5.5.5.254 eq telnet
    20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
```

ACLs dinâmicas: exercício

5. Tentar a partir de WinXP alcançar o servidor R1

```
C:\Documents and Settings\Cisco>ping 194.65.52.254
Pinging 194.65.52.254 with 32 bytes of data:
Reply from 5.5.5.254: Destination net unreachable.

Ping statistics for 194.65.52.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Documents and Settings\Cisco>telnet 5.5.5.254
```

Acesso negado!

```
c:\ Telnet 5.5.5.254
----- Router R-ISP -----
- Password: cisco

User Access Verification
Username: user1
Password:
Connection to host lost.
```

6. Autenticar-se em
RISP



Sessão terminada após autenticação



ACLs dinâmicas: exercício

7. Inspeccionar a ACL dinâmica

```
R-ISP#show access-lists
Extended IP access list 101
 10 permit tcp any host 5.5.5.254 eq telnet (132 matches)
 20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
    permit ip host 5.5.5.5 host 194.65.52.254
```

8. Repetir o passo 5 e confirmar o acesso concedido

```
C:\Documents and Settings\Cisco>ping 194.65.52.254
Pinging 194.65.52.254 with 32 bytes of data:
Reply from 194.65.52.254: bytes=32 time=130ms TTL=254
Reply from 194.65.52.254: bytes=32 time=82ms TTL=254
Reply from 194.65.52.254: bytes=32 time=52ms TTL=254
Reply from 194.65.52.254: bytes=32 time=32ms TTL=254

Ping statistics for 194.65.52.254:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 32ms, Maximum = 130ms, Average = 74ms
```

- Entrada temporária criada na ACL dinâmica em resultado da autenticação bem sucedida.
- O *template* definido (user1-list) aquando da criação da ACL dinâmica vê substituído no SA/WM o valor “host 5.5.5.5” uma vez que a ACL 101 é de ingresso e no autocmd foi especificada a opção host.

ACLs dinâmicas: exercício

9. Inspeccionar a ACL dinâmica

```
R-ISP#show access-lists
```

```
Extended IP access list 101
```

```
 10 permit tcp any host 5.5.5.254 eq telnet (132 matches)
  20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
      permit ip host 5.5.5.5 host 194.65.52.254 (4 matches) (time left 290)
```

```
...
```

```
R-ISP#sh access-lists
```

```
Extended IP access list 101
```

```
 10 permit tcp any host 5.5.5.254 eq telnet (132 matches)
  20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
      permit ip host 5.5.5.5 host 194.65.52.254 (4 matches) (time left 42)
```

```
C:\Documents and Settings\Cisco>ping 194.65.52.254
```

```
Pinging 194.65.52.254 with 32 bytes of data:
```

```
Reply from 194.65.52.254: bytes=32 time=66ms TTL=254
Reply from 194.65.52.254: bytes=32 time=47ms TTL=254
```

O *timeout* programado no *autocommand* representa o *idle timeout* da sessão. O seu *countdown* é reiniciado com a passagem do 1º pacote de cada sessão.

```
R-ISP#show access-lists
```

```
...
```

```
      permit ip host 5.5.5.5 host 194.65.52.254 (8 matches) (time left 299)
```

ACLs dinâmicas: exercício

10. Se a entrada temporária da ACL dinâmica não for estimulada por novas sessões oriundas da mesma origem acabará por desaparecer. O mesmo sucede quando o *absolute timer* programado no *template* expirar (mesmo que esteja a decorrer uma sessão).

```
R-ISP#show access-lists
R-ISP#sh access-lists
Extended IP access list 101
    10 permit tcp any host 5.5.5.254 eq telnet (132 matches)
    20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
        permit ip host 5.5.5.5 host 194.65.52.254 (4 matches) (time left 20)
...
R-ISP#show access-lists
Extended IP access list 101
    10 permit tcp any host 5.5.5.254 eq telnet (132 matches)
    20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
```



ACLs dinâmicas: exercício

11. Repetir a autenticação em RISP a partir de R1

```
R1-Firewall#telnet 5.5.5.254
Trying 5.5.5.254 ... Open

----- Router R-ISPs -----
- Password: cisco
-----
User Access Verification
Username: user1
Password:
% No input access group defined for Serial0/0.
[Connection to 5.5.5.254 closed by foreign host]
```

A interface S0/0 não se encontra programada com nenhuma ACL dinâmica de ingresso pelo que a aplicação do *autocommand* falha.



ACLs dinâmicas: exercício

12. Refazer o *autocommand* retirando a opção *host* e não especificando o *session timeout*

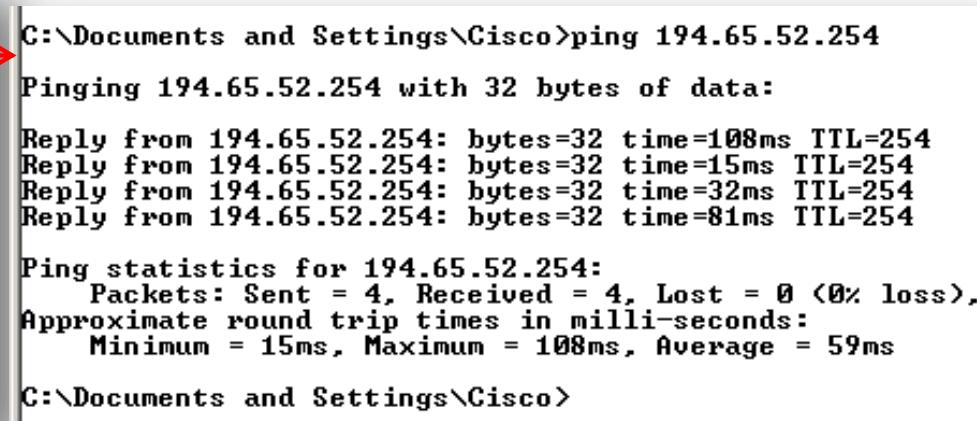
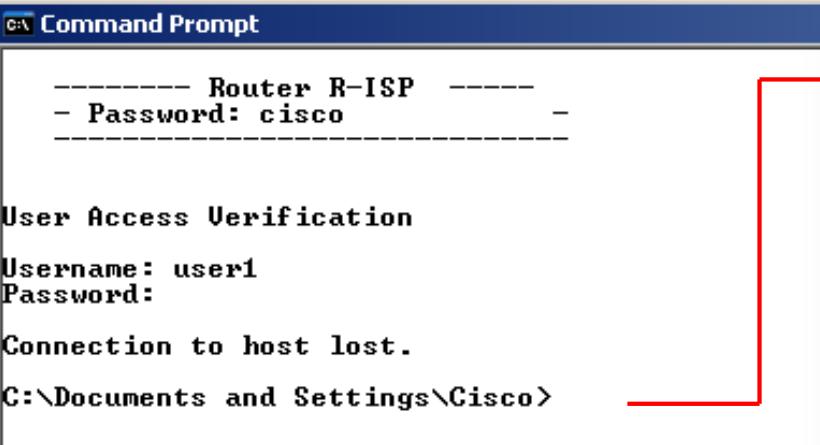
```
R-ISP#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R-ISP(config)#line vty 0 4
R-ISP(config-line)#no autocommand access-enable host timeout 5
R-ISP(config-line)#autocommand access-enable
R-ISP(config-line)#^Z

R-ISP#sh access-lists 101
Extended IP access list 101
    10 permit tcp any host 5.5.5.254 eq telnet (204 matches)
    20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
R-ISP#
```



ACLs dinâmicas: exercício

13. Efectuar a autenticação a partir de WinXP e reparar que o *session timeout* deixa de ser reportado



```
Command Prompt
-----
Router R-ISP -----
- Password: cisco - 

User Access Verification

Username: user1
Password:

Connection to host lost.

C:\Documents and Settings\Cisco>
```

```
C:\Documents and Settings\Cisco>ping 194.65.52.254
Pinging 194.65.52.254 with 32 bytes of data:
Reply from 194.65.52.254: bytes=32 time=108ms TTL=254
Reply from 194.65.52.254: bytes=32 time=15ms TTL=254
Reply from 194.65.52.254: bytes=32 time=32ms TTL=254
Reply from 194.65.52.254: bytes=32 time=81ms TTL=254

Ping statistics for 194.65.52.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 15ms, Maximum = 108ms, Average = 59ms

C:\Documents and Settings\Cisco>
```

```
R-ISP#sh access-lists 101
Extended IP access list 101
 10 permit tcp any host 5.5.5.254 eq telnet (276 matches)
 20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
      permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254 (4 matches)
```

O template é concretizado de forma diferente (a entrada temporária vem dar acesso a todos os terminais da sua rede) e o *session timeout* deixa de existir.

ACLs dinâmicas: exercício

14. Como apagar uma entrada temporária da ACL dinâmica?

```
R-ISP#clear access-template 101 user1-list 5.5.5.0 0.0.0.255 host  
194.65.52.254  
R-ISP#show access-lists 101  
Extended IP access list 101  
    10 permit tcp any host 5.5.5.254 eq telnet (591 matches)  
    20 Dynamic user1-list permit ip 5.5.5.0 0.0.0.255 host 194.65.52.254
```



ACLs dinâmicas: desafio

- Complemente o desafio das ACL reflexivas considerando o seguinte requisito adicional:
 - A partir da DMZ os administradores de R1-Firewall devem poder abrir sessões para a intranet quando devidamente autenticados naquele router.



ACLs dinâmicas: desafio

- Solução
 - TPC

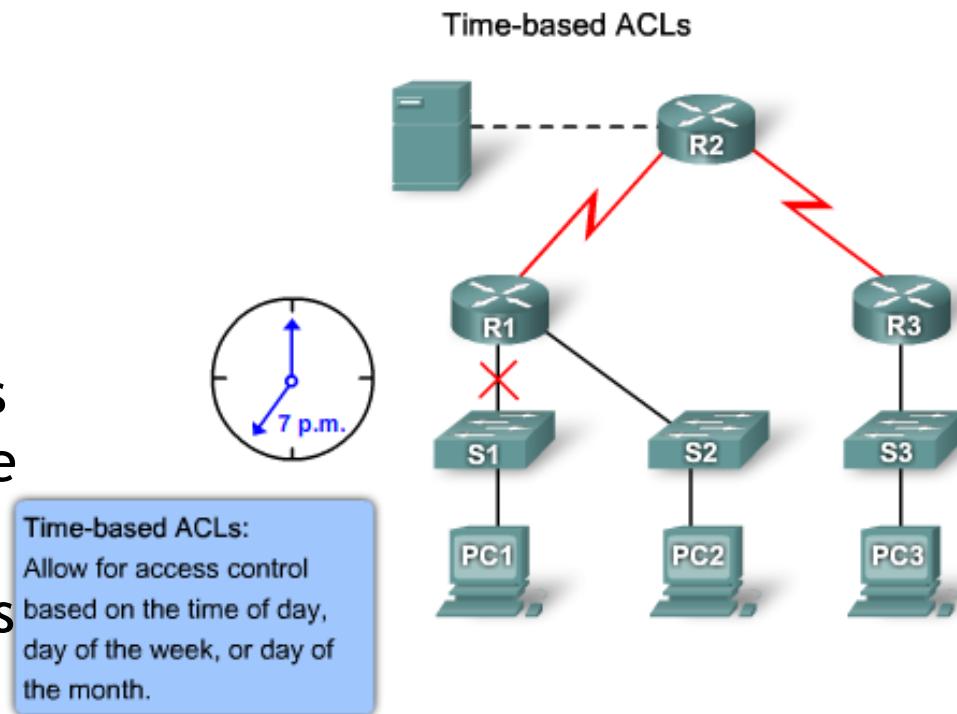


Time-based ACLs

DEIS

Time-based ACLs

- 1998: IOS > 12.0(1)T
- Permite definir o período em que cada teste das *extended ACLs (named ou numbered)* vigora:
 - Hora do dia
 - Dia da semana
 - Dia do mês
- Exemplos:
 - Permitir que os funcionários acedam à Internet à hora de almoço
 - Gerar *logs* apenas em certos período do dia



Time-based ACLs: configuração

1. Definir o período temporal em que deve vigorar a ACL

O nome deve começar por um carácter alfabético. Sugestão: usar maiúsculas em todos os identificadores definidos pelo programador do router.

```
Router(config) # time-range time_range_name  
Router(config-time-range) # absolute [start_time start_date] [end_time end_date]
```

23:59 → Por omissão é assumida a hora e dia actuais
31 May 2010 ↑
Expira no início do minuto seguinte

```
Router(config-time-range) # periodic day_of_the_week hh:mm to  
[day_of_the_week] hh:mm
```

Friday Monday Saturday Sunday
Thursday Tuesday Wednesday
daily Every day of the week
weekdays Monday thru Friday
weekend Saturday and Sunday
Podem ainda ser indicados vários dias da semana. Ex.: Monday Sunday

Por omissão é assumido o mesmo dia de início do período.

Por omissão é assumido 23:59 31 December 2035.

Time-based ACLs: configuração

2. Num ou em vários testes de uma ACL (*numbered* ou *named*) activar (i.e., aplicar) o *time range* criado

```
Router(config)# access-list {100-199} {permit | deny} protocol source-  
addr [source-mask] [operator operand] destination-addr [destination-mask]  
[operator operand] [established] [log | log-input] [established] [time-range  
name_of_time_range]
```

- Nota: Na mesma ACL podem ser definidos testes com e sem *time range*.
- Nota: Na mesma ACL podem ser definidos em testes distintos *time ranges* distintos.



Time-based ACLs: Exemplo

- Permitir que os empregados da intranet 192.168.10.0/24 efectuem acessos *telnet* ao exterior segundas, terças e sextas durante o horário laboral (8h00-17h00).

Step 1

```
R1(config)# time-range EVERYOTHERDAY  
R1(config-time-range)# periodic Monday Wednesday Friday 8:00 to  
17:00
```

Step 2

```
R1(config)# access-list 101 permit tcp 192.168.10.0 0.0.0.255  
any eq telnet time-range EVERYOTHERDAY
```

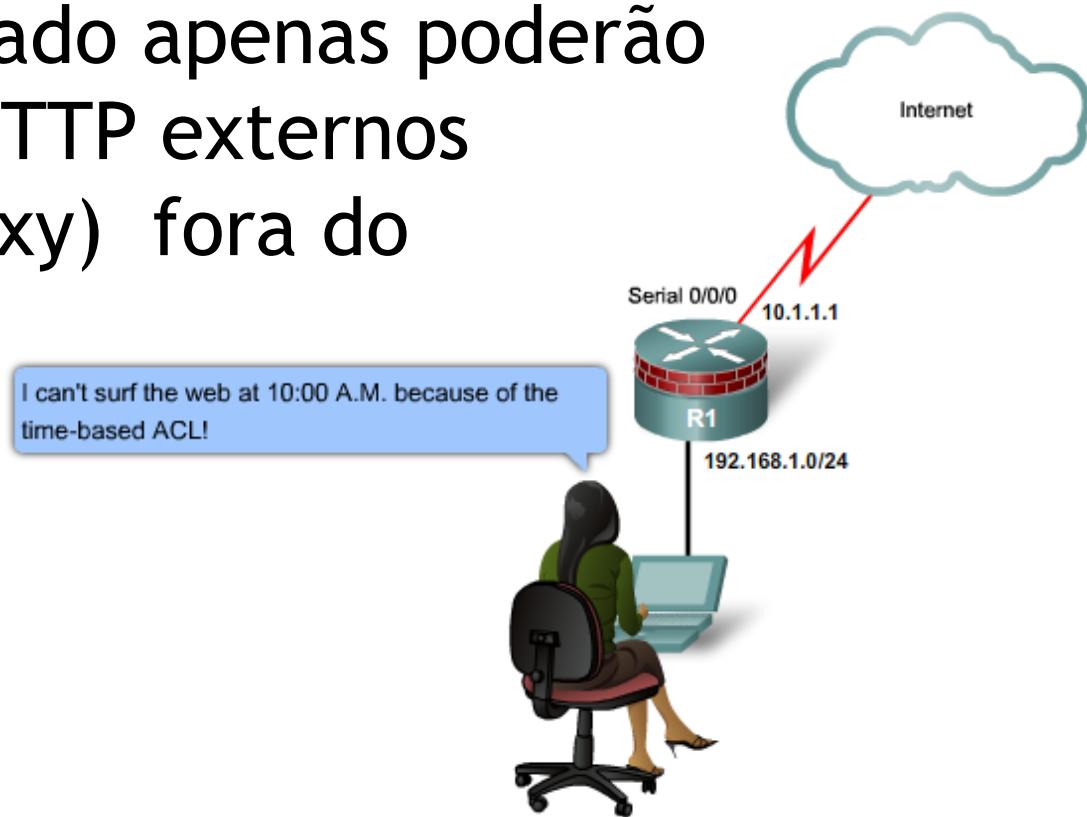
Step 3

```
R1(config)# interface s0/0/0  
R1(config-if)# ip access-group 101 out
```



Time-based ACLs: exercício

- Desenvolva uma ACL que permita aplicar a seguinte política “os terminais da Intranet servidos por IP privado apenas poderão aceder a serviços HTTP externos (mediados pelo proxy) fora do período laboral (2^a-6^a feira / 9h00-12h30 ∪ 14h00-17h30)



Time-based ACLs: solução

1. Consulte e acerte o relógio de R1-Firewall

```
R1-Firewall#show clock
*00:02:02.499 UTC Fri Mar 1 2002
R1-Firewall#clock set 23:26:30 31 may 2012
R1-Firewall#show clock
23:26:35.023 UTC Tue May 31 2012
```

2. Defina o *time range* apropriado

```
R1-Firewall#configuration terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1-Firewall(config)#time-range PERIODO_LABORAL
R1-Firewall(config-time-range)#periodic weekdays 9:00 to 12:29
R1-Firewall(config-time-range)#periodic weekdays 14:00 to 17:29
R1-Firewall(config-time-range)#^Z
R1-Firewall#show time-range
time-range entry: PERIODO_LABORAL (inactive)
    periodic weekdays 9:00 to 12:29
    periodic weekdays 14:00 to 17:29
```



Time-based ACLs: solução

3. Defina uma ACL para o efeito e active o time range

```
R1-Firewall(config)#ip access-list extended INTRANET_DMZ-INTERNET
R1-Firewall(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 host
194.65.52.6 eq 8080 time-range PERIODO_LABORAL
R1-Firewall(config-ext-nacl)#permit ip any any
R1-Firewall(config-ext-nacl)#exit
R1-Firewall(config)#interface f0/0
R1-Firewall(config-if)#ip access-group INTRANET_DMZ-INTERNET in
R1-Firewall(config-if)#^z
```

- Nota: Os acessos são feitos através do proxy da DMZ

4. Active a depuração de pacotes IP em R1-Firewall

```
R1-Firewall#debug ip packet
IP packet debugging is on
R1-Firewall#
```



Time-based ACLs: solução

5. Teste o acesso TCP a partir de C1 ao proxy (194.65.52.6:8080) e verifique o comportamento do tráfego face à hora presente

```
VPCS[4]> 1
VPCS[1]> ping 194.65.52.6 -3 -1 100 -p 8080
Connect 8080@194.65.52.6 timeout
...
```

```
R1-Firewall#
Jun  1 00:59:12.003: IP: tableid=0, s=192.168.1.1 (FastEthernet0/0),
d=194.65.52.6 (FastEthernet0/1), routed via RIB
Jun  1 00:59:12.007: IP: s=192.168.1.1 (FastEthernet0/0), d=194.65.52.6
(FastEthernet0/1), g=194.65.52.6, len 140, forward
Jun  1 00:59:12.015: IP: s=192.168.1.1 (FastEthernet0/0), d=194.65.52.6
(FastEthernet0/1), len 140, encapsulation failed
R1-Firewall#
```



Time-based ACLs: solução

6. Altere a hora actual de R1-Firewall de modo a que este inverta o tratamento que dá ao tráfego dirigido ao proxy pelos terminais privados da intranet

```
R1-Firewall#clock set 9:30:00 1 June 2012
```

7. Teste de novo a resposta que o tráfego de C1 tem

```
VPCS[1]> ping 194.65.52.6 -3 -1 100 -p 8080  
Connect 8080@194.65.52.6 timeout
```

```
Jun 1 09:30:28.163: IP: s=192.168.1.1 (FastEthernet0/0), d=194.65.52.6,  
len 140, access denied  
R1-Firewall#sh access-lists  
Extended IP access list INTRANET_DMZ-INTERNET  
 10 deny tcp 192.168.1.0 0.0.0.255 host 194.65.52.6 eq 8080 time-range  
PERIODO_LABORAL (active) (15 matches)  
 20 permit ip any any (15 matches)
```



Troubleshooting de ACLs

DEIS

Comandos de apoio à depuração de ACLs

- Consultar as ACL instaladas e a cobertura dos seus testes perante o tráfego da rede

```
Router# show access-lists [access-list-number | access-list-name]
```

- Analisar como determinado tipo de pacotes é processado assim que entra no *router*

```
Router# debug ip packet [access-list-number] [detail]
```

- Nota: Este comando não deve ser usado em *routers* em produção uma vez que pode consumir muitos recursos (CPU/RAM/tempo de consola). Nesse cenário devem inspecionar-se os contadores da ACL, usar a opção *log* ou usar o wireshark + *port mirroring*
- Nota: Apenas podem ser usadas *numbered ACLs* com este comando (para limitar informação sobre pacotes inspecionados)



Comandos de apoio à depuração de ACLs

```
Perimeter# show access-list 100
Extended IP access list 100
permit tcp any host 200.1.1.14 eq www (189 matches) ←
permit udp any host 200.1.1.13 eq domain (32 matches)
permit tcp any host 200.1.1.12 eq smtp
permit tcp any eq smtp host 200.1.1.12 established
permit tcp any host 200.1.1.11 eq ftp
permit tcp any host 200.1.1.11 eq ftp-data
permit tcp any eq www 200.1.2.0 0.0.0.255 established
permit udp any eq domain 200.1.2.0 0.0.0.255
deny ip any any (1237 matches)
```

Os contadores dão informação preciosa mas não mostram com detalhe as características do tráfego classificado

```
Edge# debug ip packet
IP packet debugging is on
```

```
IP: s-200.0.2.2 (FastEthernet0/0), d-172.69.2.42 (Serial0/0/0), g-172.69.16.2, forward
IP: s-200.0.2.2 (FastEthernet0/0), d-172.16.2.42 (Serial0/0/0), g-172.69.16.2, forward
IP: s-200.0.2.2 (FastEthernet0/0), d-172.69.43.126 (Serial0/0/0), g-172.69.16.2, forward
IP: s-200.0.2.2 (FastEthernet0/0), d-172.16.2.42 (Serial0/0/0), g-172.69.16.2, access denied
```

É indicado o próximo gateway do pacote

Mitigação de ataques com ACLs

DEIS

Comandos de apoio à depuração de ACLs

- Ataques que podem ser mitigados com recurso a ACLs
 - IP Spoofing (inbound/outbound)
 - Falsear o endereço fonte de pacotes IP
 - Ataques DoS TCP SYN
 - Abertura incompleta e massiva de sessões TCP de modo a saturar os servidores e impedir que clientes legítimos possuam
 - Ataques DoS smurf
 - Deixar entrar numa rede pacotes ilegítimos destinados a endereços de broadcast/multicast.
 - Mensagens ICMP perigosas
 - Actividades de inspecção topológica (*traceroute*)



Comandos de apoio à depuração de ACLs

- Relativamente ao IP *spoofing*
 - Deve-se exportar tráfego apenas do nosso espaço público
 - Evitamos alojar sistemas que perpetrem este tipo de ataques
 - Não se deve importar tráfego com endereços inaceitáveis.
 - Rob Thomas mostrou que 60% ataques usam estes endereços
 - Endereços de *loopback*: 127.0.0.0/8
 - Endereços privados: RFC 1918
 - Endereços *multicast*: 224
 - <http://www.team-cymru.org/Services/Bogons/>
 - Políticas mais agressivas
 - <http://www.ris.ripe.net/debogon/>

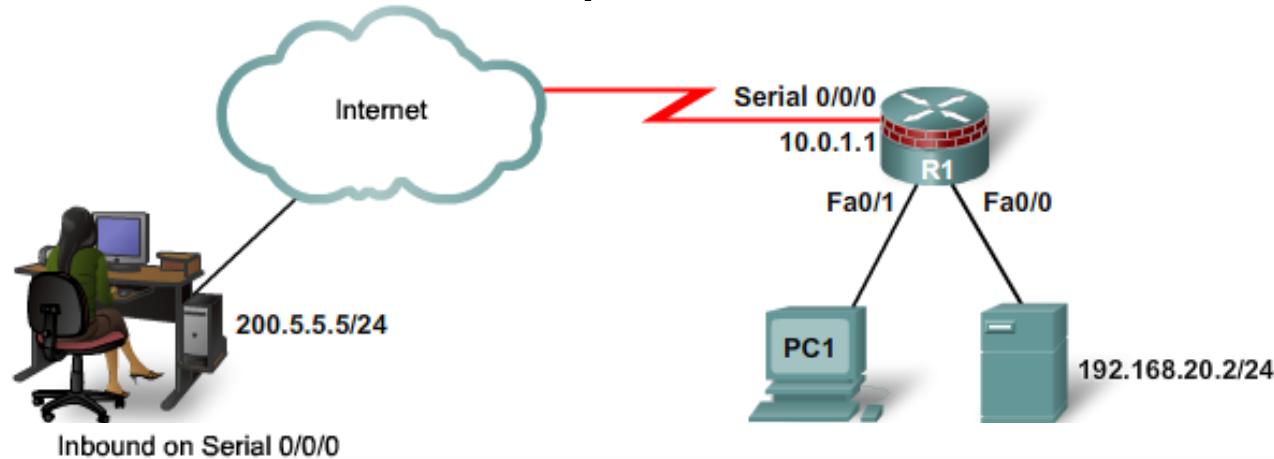
0.0.0.0 0.255.255.255
10.0.0.0 0.255.255.255
127.0.0.0 0.255.255.255
169.254.0.0 0.0.255.255
172.16.0.0 0.15.255.255
192.0.0.0 0.0.0.255
192.0.2.0 0.0.0.255
192.168.0.0 0.0.255.255
198.18.0.0 0.1.255.255
198.51.100.0 0.0.0.255
203.0.113.0 0.0.0.255
224.0.0.0 31.255.255.255



<http://www.cymru.com/Documents/secure-ios-template.html>

Comandos de apoio à depuração de ACLs

- No entanto as redes não podem viver isoladas
 - Exemplo: muitas vezes é estritamente necessário administrar um *router* a partir da Internet



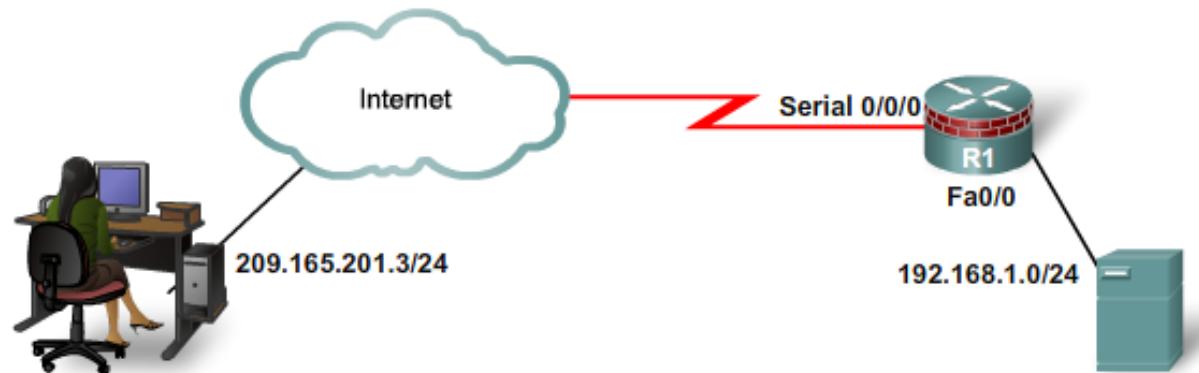
Inbound on Serial 0/0/0

```
R1(config)# access-list 180 permit udp any host 192.168.20.2 eq domain
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq smtp
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq ftp
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 10.0.1.1 eq telnet
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 10.0.1.1 eq 22
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq syslog
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq snmptrap
```



Comandos de apoio à depuração de ACLs

- Limitação de tráfego ICMP ao mínimo indispensável



Tráfego *inbound*
indispensável →

Inbound on S0/0/0

```
R1 (config) # access-list 112 permit icmp any any echo-reply
R1 (config) # access-list 112 permit icmp any any source-quench
R1 (config) # access-list 112 permit icmp any any unreachable
R1 (config) # access-list 112 deny icmp any any
R1 (config) # access-list 112 permit ip any any
```

Tráfego *outbound*
indispensável →

Inbound on Fa0/0

```
R1 (config) # access-list 114 permit icmp 192.168.1.0 0.0.0.255 any echo
R1 (config) # access-list 114 permit icmp 192.168.1.0 0.0.0.255 any parameter-problem
R1 (config) # access-list 114 permit icmp 192.168.1.0 0.0.0.255 any packet-too-big
R1 (config) # access-list 114 permit icmp 192.168.1.0 0.0.0.255 any source-quench
R1 (config) # access-list 114 deny icmp any any
R1 (config) # access-list 114 permit ip any any
```

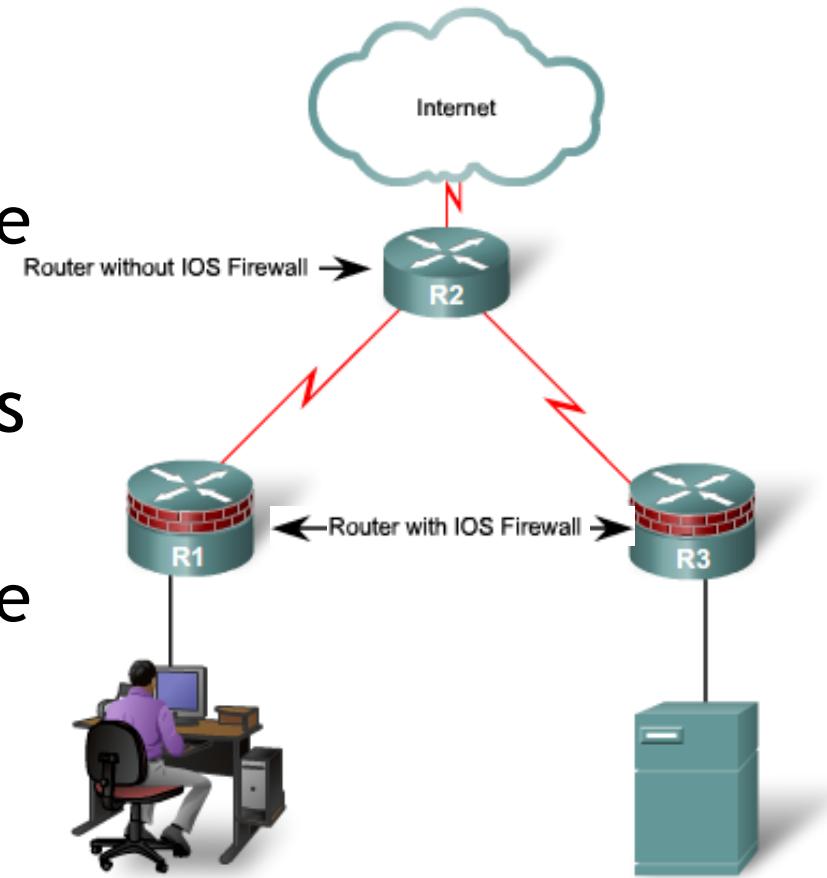


Tecnologias de *Firewalls*

DEIS

Evolução tecnológica das Firewalls

- 1988: DEC criou a primeira *firewall stateless*:
 - O veredito sobre um pacote depende apenas do mesmo
- 1989: AT&T Bell Laboratories criou a 1ª *firewall statefull*:
 - O veredito sobre um pacote depende do mesmo e, caso existam, dos anteriores que pertencem à mesma sessão



Evolução tecnológica das Firewalls

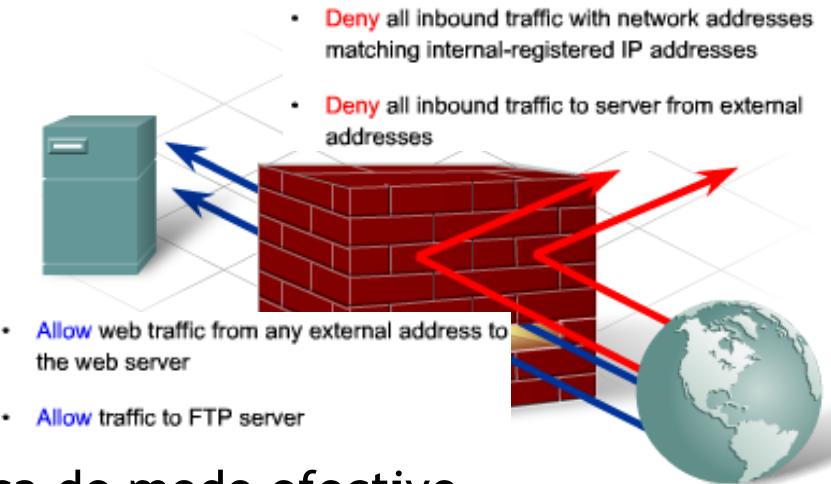
- As *firewalls* originais representavam uma funcionalidade adicional de *routers* ou servidores:
 - Resistentes a ataques
 - Pontos de trânsito de tráfego entre redes
 - Usadas para aplicar políticas de segurança
- Com funcionalidades cada vez mais complexas as *firewalls* foram ganhando lugar como dispositivos próprios de modo a “aliviar” *routers* e servidores da carga de processamento
- Em redes de dimensão moderada assiste-se hoje à integração da funcionalidade de *firewall* em *routers* de acesso
 - Ex.: Cisco Integrated Services Routers (ISRs)



Prós e contras das Firewalls

- Vantagens

- Protege de exposições perigosas sistemas, aplicações e pessoas
- Permite disciplinar a operação dos protocolos
- Dados maliciosos podem ser descartados
- Permite aplicar políticas de segurança de modo efectivo
- Concentra em poucos pontos os esforços de gestão da segurança

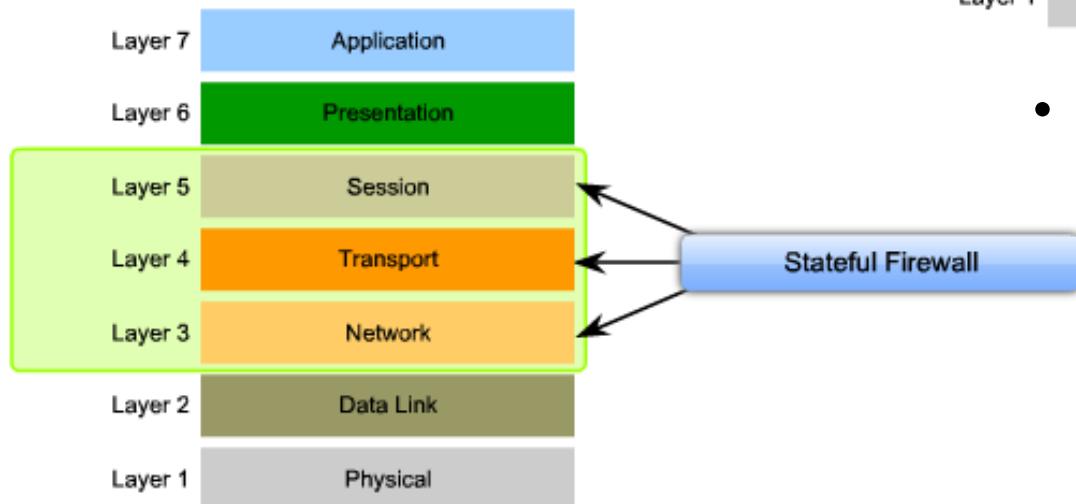
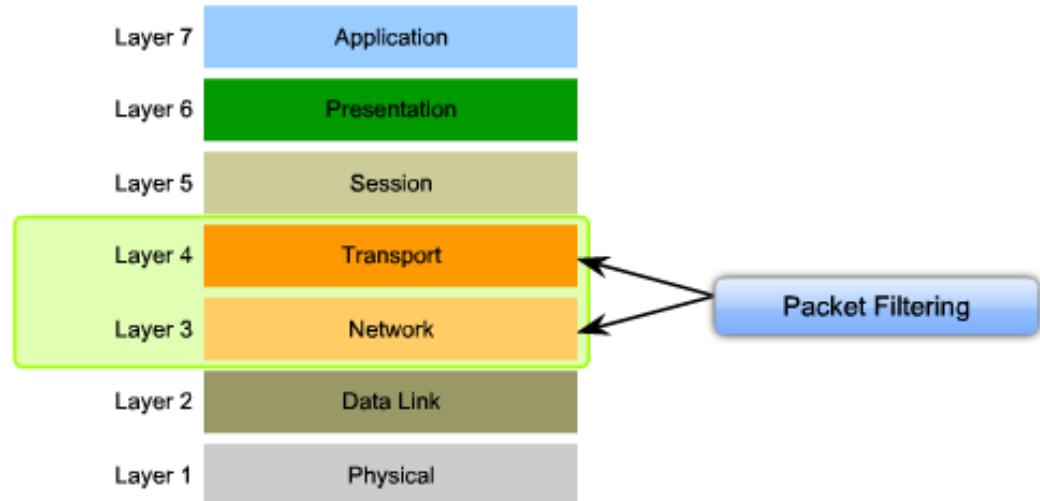


- Desvantagens

- Representa um ponto crítico de falha (problemático se mal configurada)
- Não é possível em muitas aplicações um controlo de segurança efectivo
- Os utilizadores podem contornar a *firewall* e por toda a rede em perigo
- Podes tornar-se um estrangulamento à rede prejudicando o desempenho

Tipos de Firewalls

- *Packet-filtering firewall*
 - Tipicamente implementadas como funcionalidades dos routers e capazes de tomar decisões com base em informação de nível 3 (rede) e 4 (transporte)

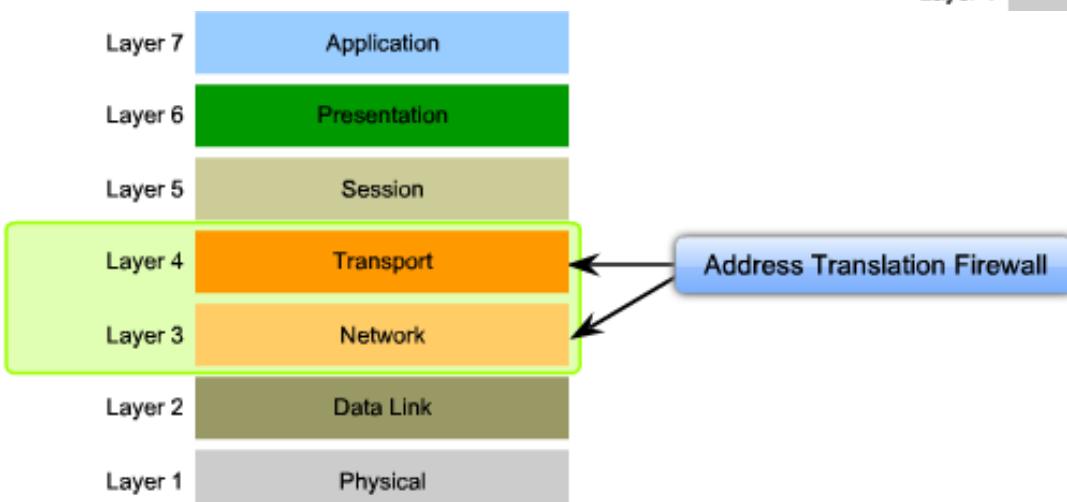
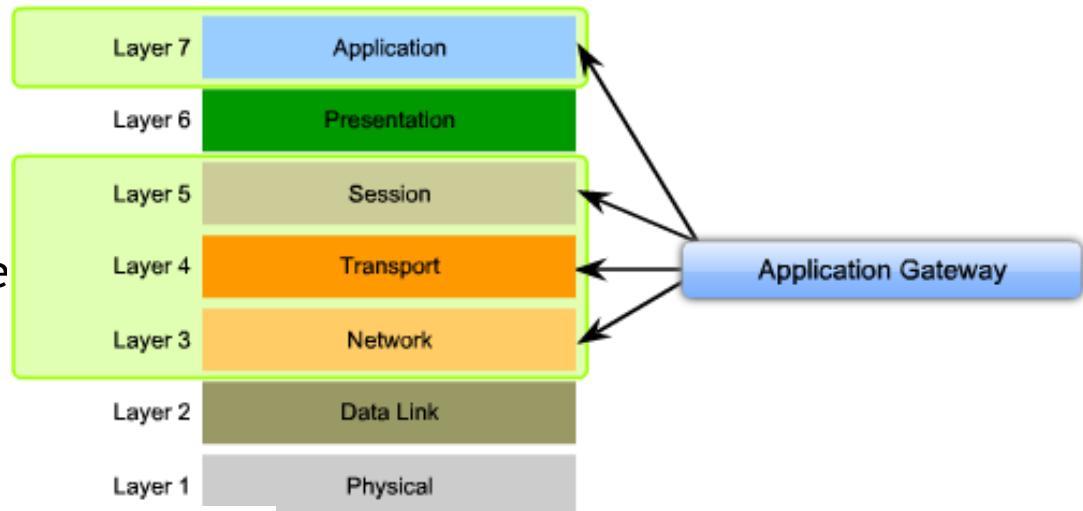


- *Stateful firewall*

- Capazes de monitorar o estado de cada sessão acompanhando a vida destas desde o estabelecimento, passando pela transferência de dados, até ao seu encerramento

Tipos de Firewalls

- *Application gateway firewall (proxy firewall)*
 - Possuem inteligência capaz de estender a inspecção de tráfego até ao nível de detalhe das próprias aplicações.
 - Tipicamente implementadas em software



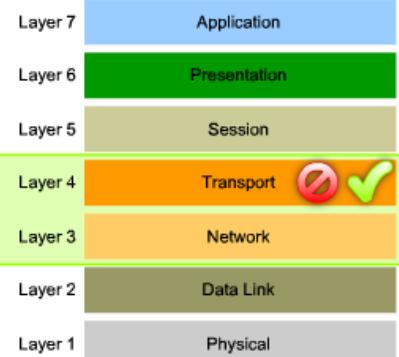
- *Address-translation firewall*
 - Possuem adicionalmente a capacidade de efectuar translação de endereços (NAT)

Tipos de Firewalls

- Outros tipos de firewalls
 - *Transparent firewall*
 - Firewalls que possuem capacidade de efectuar vigilância e disciplina sobre tráfego entre portas *bridged* (i.e. ao nível 2)
 - *Host-based firewall* (servidor ou pessoal)
 - Aplicação ou serviço de software capaz de vigiar sessões iniciadas e terminadas localmente (i.e., num servidor / desktop / laptop).
 - *Hybrid firewall*
 - Combina características de vários tipos de firewalls



Packet-filtering firewall



- Simple permit or deny rule sets can be used to implement a packet filter.
- Packet filters have a low impact on network performance.
- Packet filters are easy to implement, and are supported by most routers.
- An initial degree of security at a low network layer can be provided by a packet filter.
- A packet filter can perform almost all tasks of a high end firewall at a much lower cost.
- Packet filtering is susceptible to IP spoofing. Hackers send arbitrary packets that fit ACL criteria and pass through the filter.
- Packet filters do not filter fragmented packets well. Because fragmented IP packets carry the TCP header in the first fragment and packet filters filter on TCP header information, all fragments after the first fragment are passed unconditionally. Decisions to use packet filters assume that the filter of the first fragment accurately enforces the policy.
- Complex ACLs are difficult to implement and maintain correctly.
- Packet filters cannot dynamically filter certain services. For example, sessions that use dynamic port negotiations are difficult to filter without opening access to a whole range of ports.
- Packet filters are stateless. They examine each packet individually rather than in the context of the state of a connection.



Statefull firewall

The diagram illustrates the Stateful Firewall architecture. On the left, the OSI model layers are shown from Layer 1 (Physical) to Layer 7 (Application). Layer 4 (Transport) is highlighted with a red 'no' symbol and a green checkmark, indicating it is monitored by the Stateful Inspection process. To the right, a network connection between the Internet and a computer passes through a Stateful Firewall. The firewall contains 'Stateful Session Flow Tables'. Below the firewall, two hosts (IP 10.1.1.1 and 209.165.201.3) are connected, with traffic flowing between them on source port 1500 to destination port 80. At the bottom, two ACL tables are shown:

Inside ACL (Outgoing Traffic)	Outside ACL (Incoming Traffic)
permit ip 10.0.0.0 0.0.0.255 any	Dynamic: permit tcp host 209.165.201.3 eq 80 host 10.1.1.1 eq 1500
	permit tcp any host 10.1.1.2 eq 25
	permit udp any host 10.1.1.2 eq 53
	deny ip any any

- Muitas possuem a capacidade de seguir o número de sequência dos segmentos TCP, perceber como opera o FTP, o DNS, ... evitando os ataques TCP RST flood, DNS cache poisoning, etc.
- Representam as *firewalls* mais versáteis e comuns actualmente
- O estado de cada sessão é mantido internamente

Statefull firewall

Advantages

- Stateful firewalls are often used as a primary means of defense by filtering unwanted, unnecessary, or undesirable traffic.
- By providing more stringent control over security, stateful firewalls strengthen packet filtering.
- Stateful firewalls improve performance over packet filters or proxy servers.
- Stateful firewalls defend against spoofing and DoS attacks by determining whether packets belong to an existing connection or are from an unauthorized source.
- More log information is provided by a stateful firewall than a packet filtering firewall.

Disadvantages

- Stateful firewalls cannot prevent Application Layer attacks because they do not examine the actual contents of the HTTP connection.
- Not all protocols are stateful, such as UDP and ICMP, these protocols do not get as much support.
- Some applications open multiple connections requiring a whole new range of ports opened to allow this second connection.
- Stateful firewalls do not support user authentication.



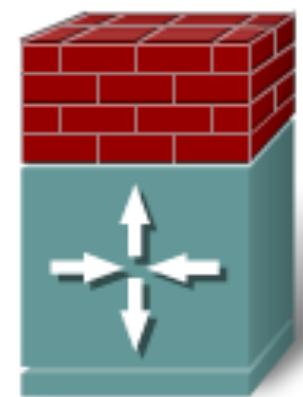
Mercado de firewalls

- Cisco IOS Firewall

- Capacidade específica do IOS destinada a suprir as necessidades de pequenas e médias redes (*small and medium-sized business - SMB*)

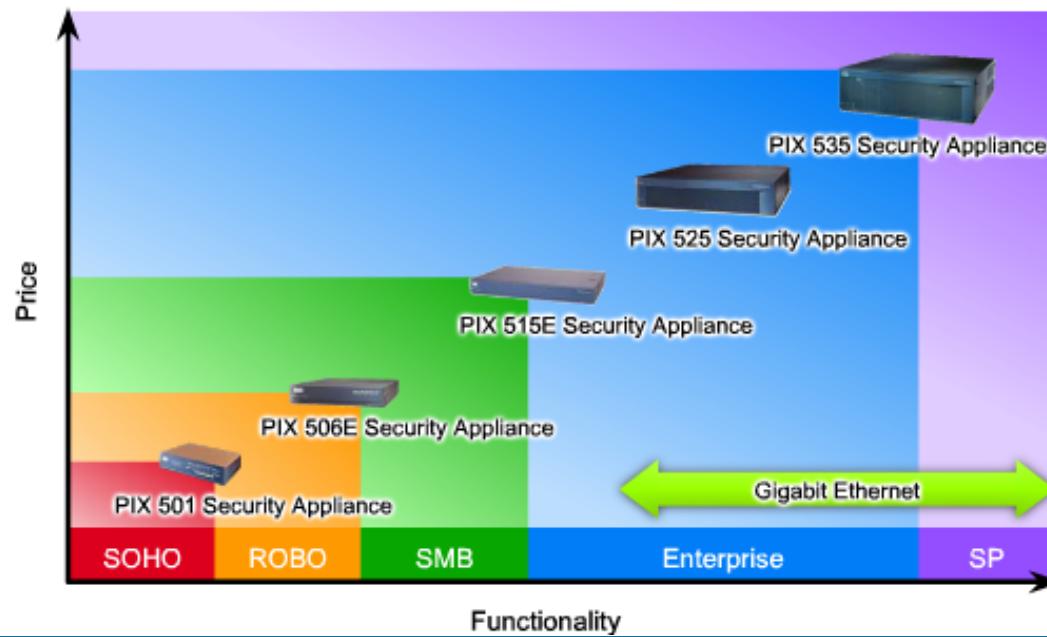
Cisco IOS Firewall Features

- Zone-based policy framework for intuitive management
- Instant messenger and peer-to-peer application filtering
- VoIP protocol firewalling
- Virtual routing and forwarding (VRF) firewalling
- Wireless integration
- Stateful failover
- Local URL whitelist and blacklist support
- Application inspection for web and email traffic



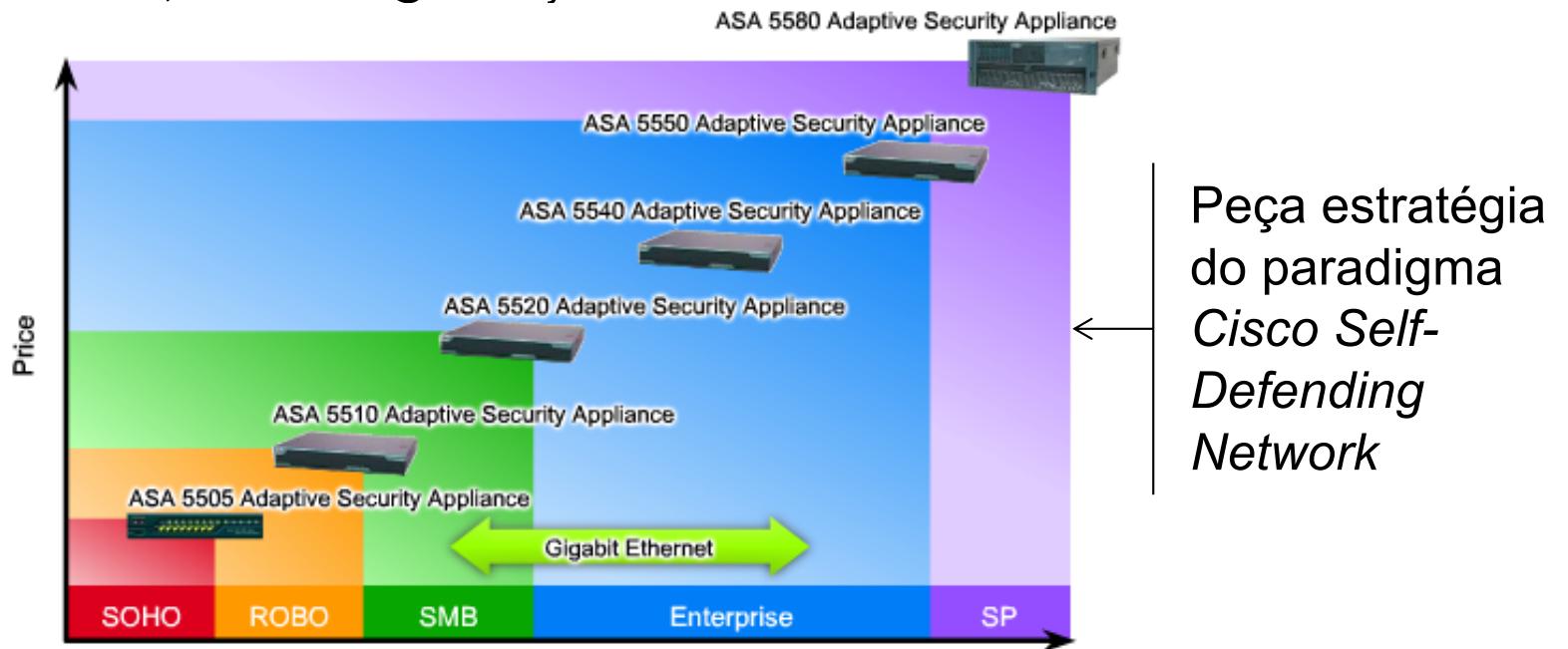
Mercado de firewalls

- Cisco PIX Security Appliance
 - Solução *standalone* e escalável capaz de forçar políticas de segurança ao nível do utilizador e aplicação, com protecção multi-vector e serviços de conectividade seguros



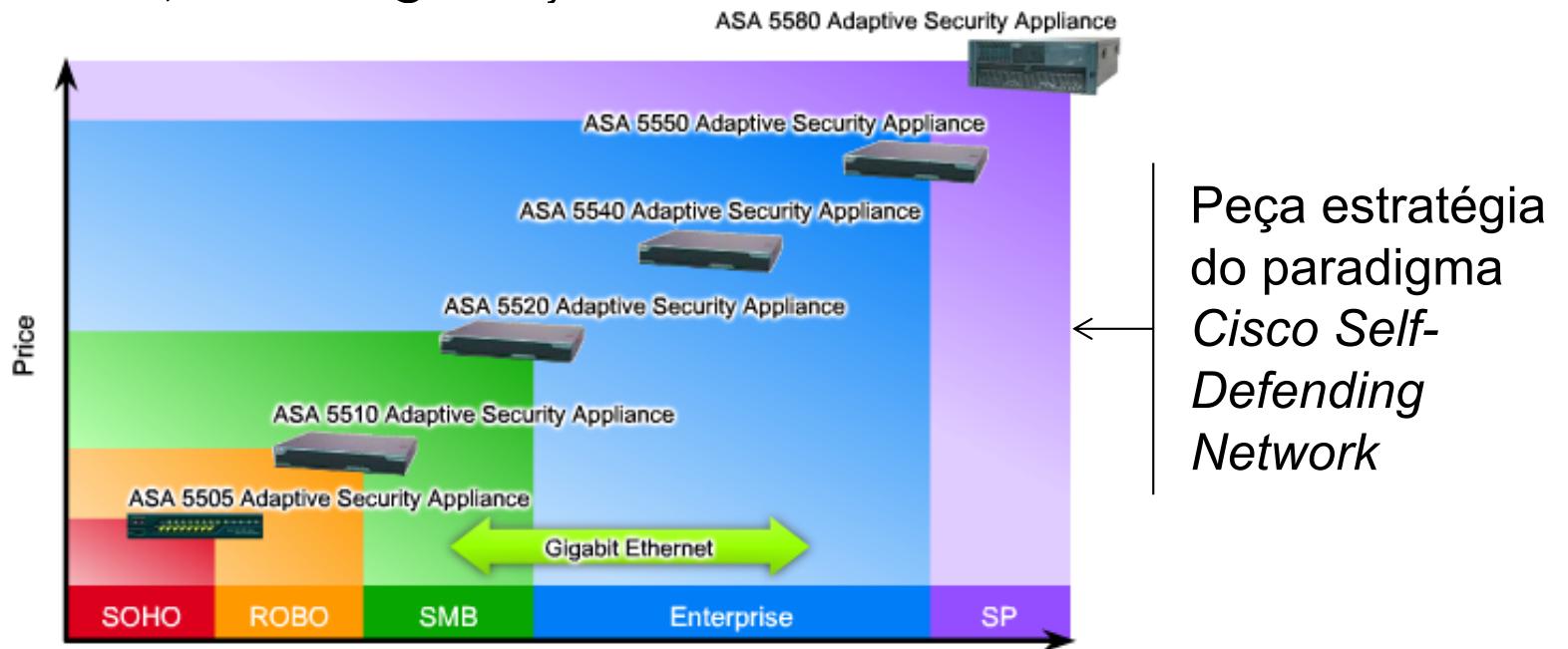
Mercado de firewalls

- *Cisco ASA Adaptive Security Appliances*
 - Solução *standalone* e escalável, simples de adaptar e que dá resposta às necessidades de segurança de voz e vídeo (Cisco Unified Communications), suporte de Secure Socket Layer (SSL), IPSec VPNs, IPS e segurança de conteúdos



Mercado de firewalls

- *Cisco ASA Adaptive Security Appliances*
 - Solução *standalone* e escalável, simples de adaptar e que dá resposta às necessidades de segurança de voz e vídeo (Cisco Unified Communications), suporte de Secure Socket Layer (SSL), IPSec VPNs, IPS e segurança de conteúdos

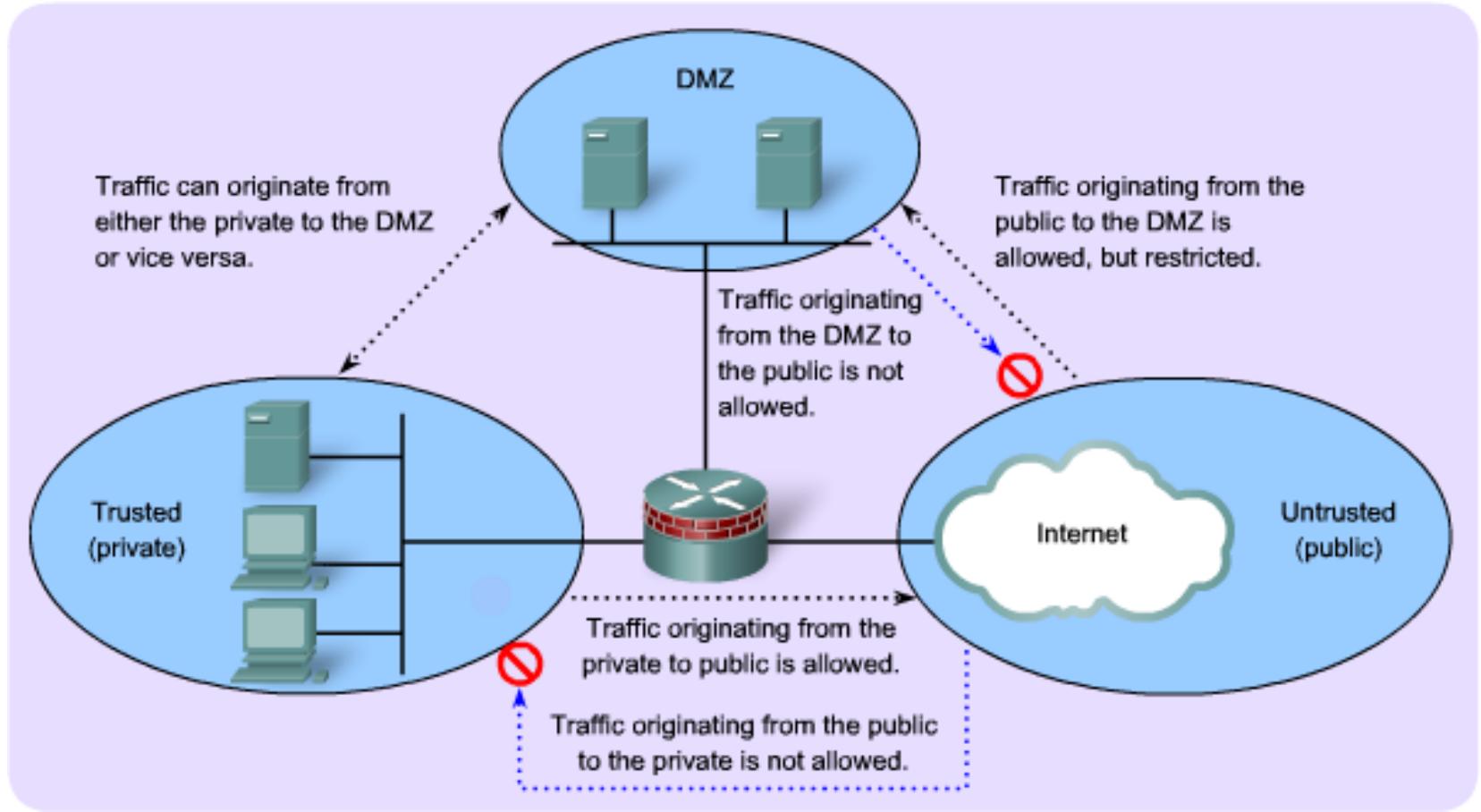


Firewalls na arquitectura da rede

DEIS

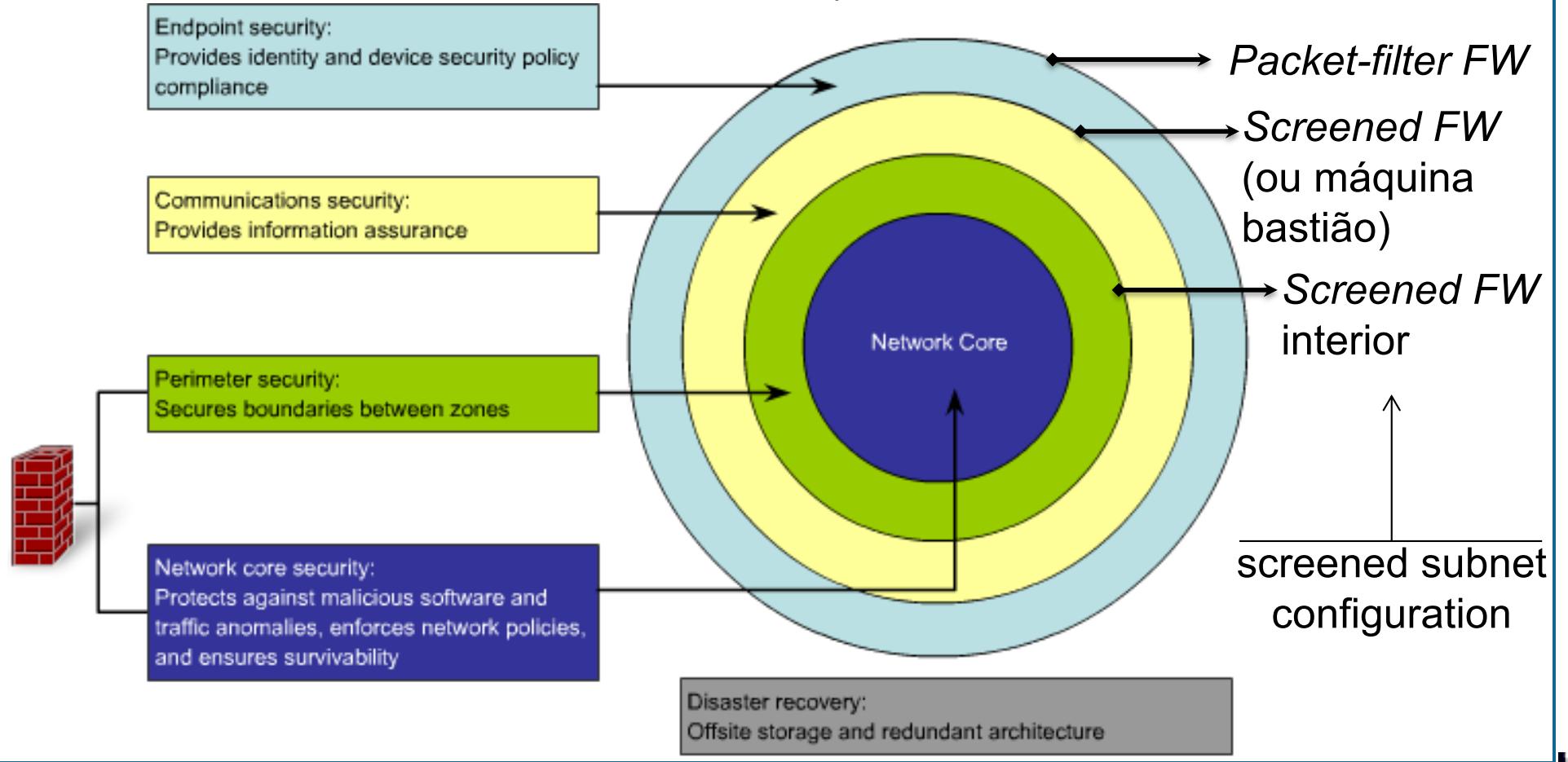
Posicionamento das *firewalls* na topologia da rede

- Zona desmilitarizada (*demilitarized zone - DMZ*)



Posicionamento das *firewalls* na topologia da rede

- Anéis sucessivos de segurança (*layered defense*)



Posicionamento das *firewalls* na topologia da rede

- Boas práticas

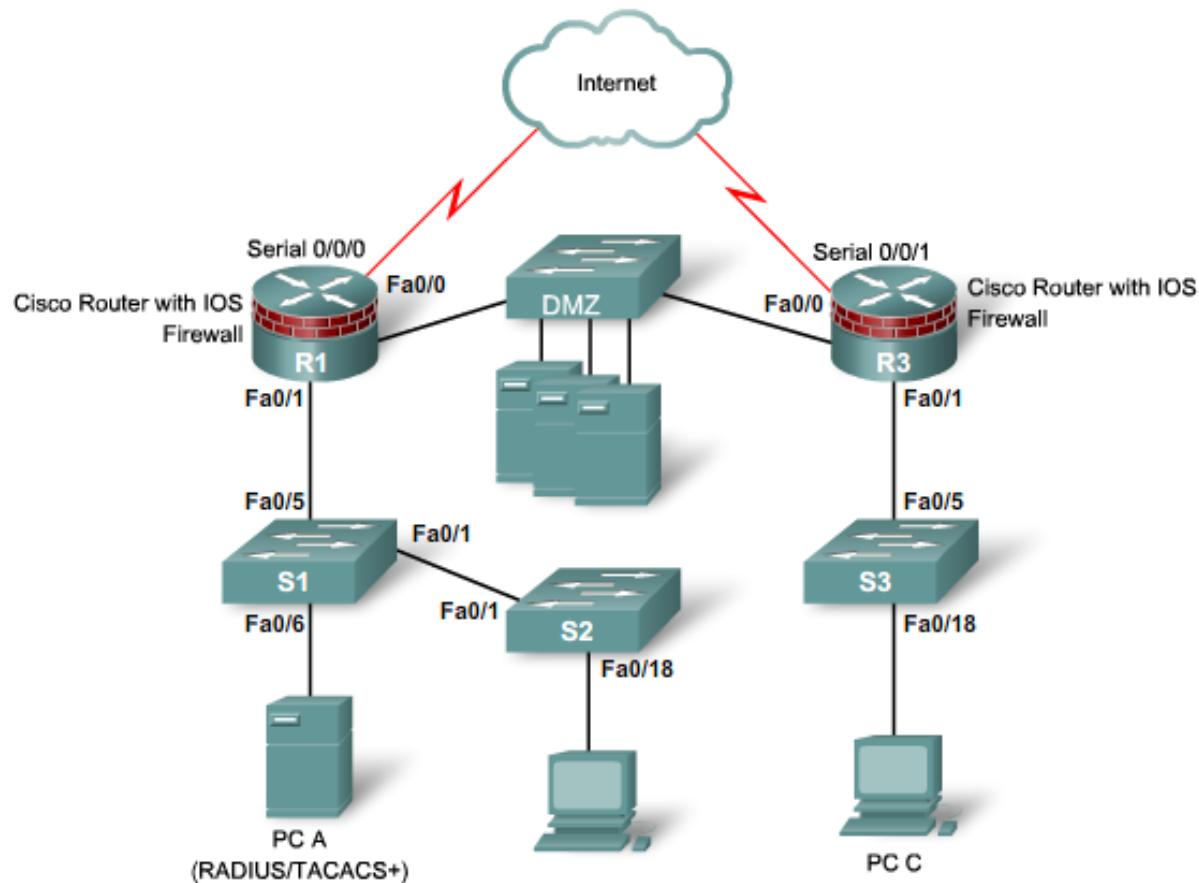
Firewall Best Practices

- Position firewalls at security boundaries.
- Firewalls are the primary security device. It is unwise to rely exclusively on a firewall for security.
- Deny all traffic by default. Permit only services that are needed.
- Ensure that physical access to the firewall is controlled.
- Regularly monitor firewall logs.
- Practice change management for firewall configuration changes.
- Remember that firewalls primarily protect from technical attacks originating from the outside.



Posicionamento das *firewalls* na topologia da rede

- Boas práticas

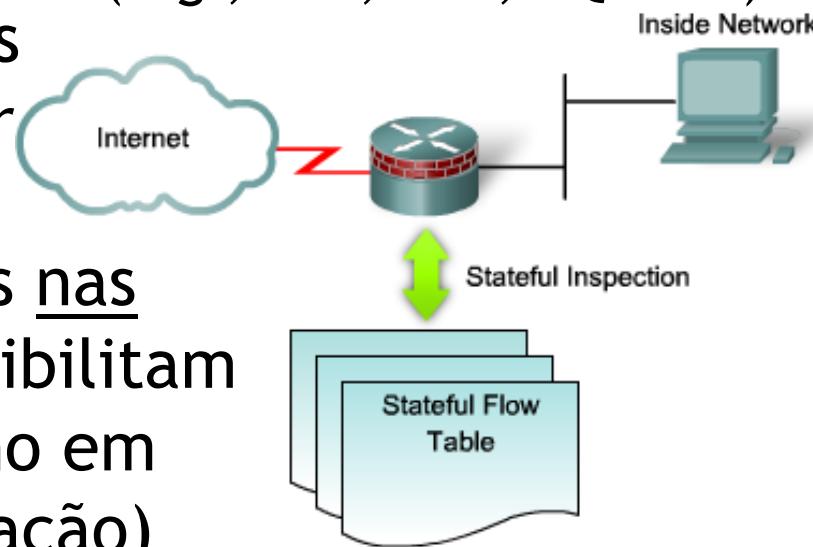


Content-based *Access Control* (CBAC)

DEIS

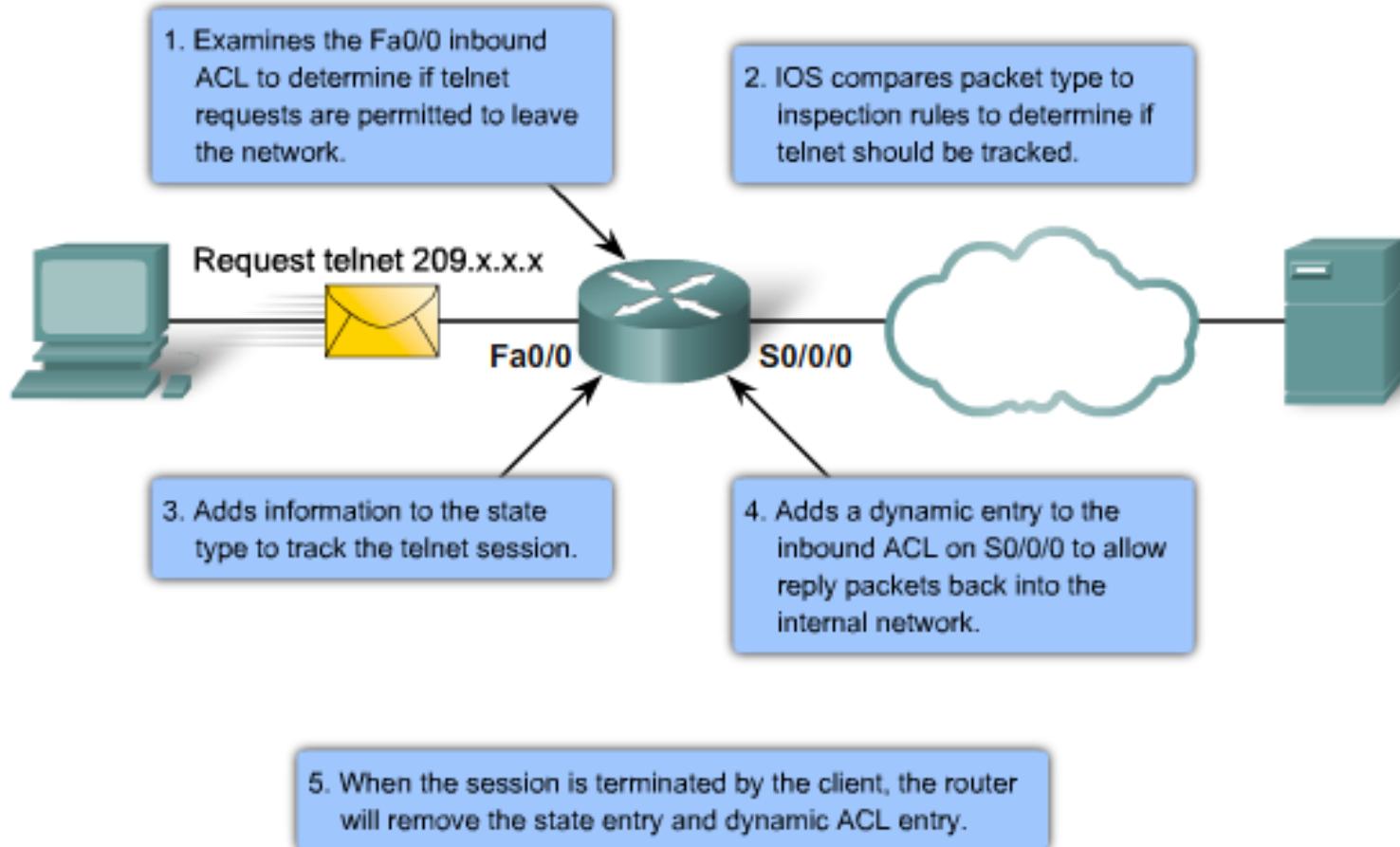
CBAC - Operação

- O controlo de acesso baseado no conteúdo (*Content-based Access Control*) é um mecanismo de *firewall stateful* introduzido no IOS 11.2 (1997)
 - O tráfego passou a ser inspeccionado ao nível da aplicação, sendo mantida informação sobre cada sessão, processada de acordo com as especificidades de cada protocolo (e.g., FTP, RPC, SQL*Net)
- A tabela de estado dos diversos fluxos que atravessam o *router* é usada para criar de forma dinâmica entradas temporárias nas interfaces adequadas que possibilitam o trânsito do tráfego de retorno em moldes legítimos (para a aplicação)



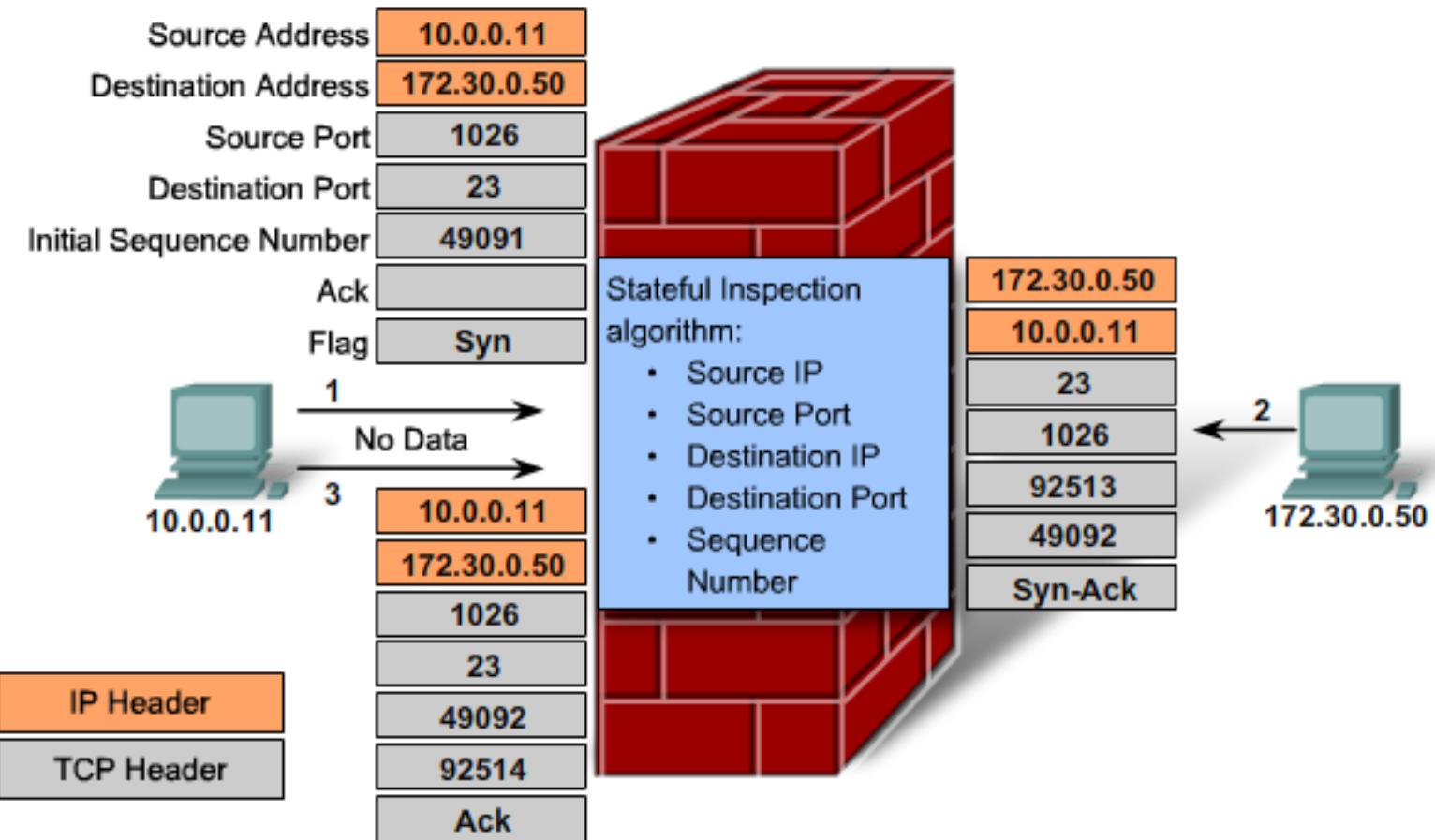
CBAC - Operação

- Exemplo



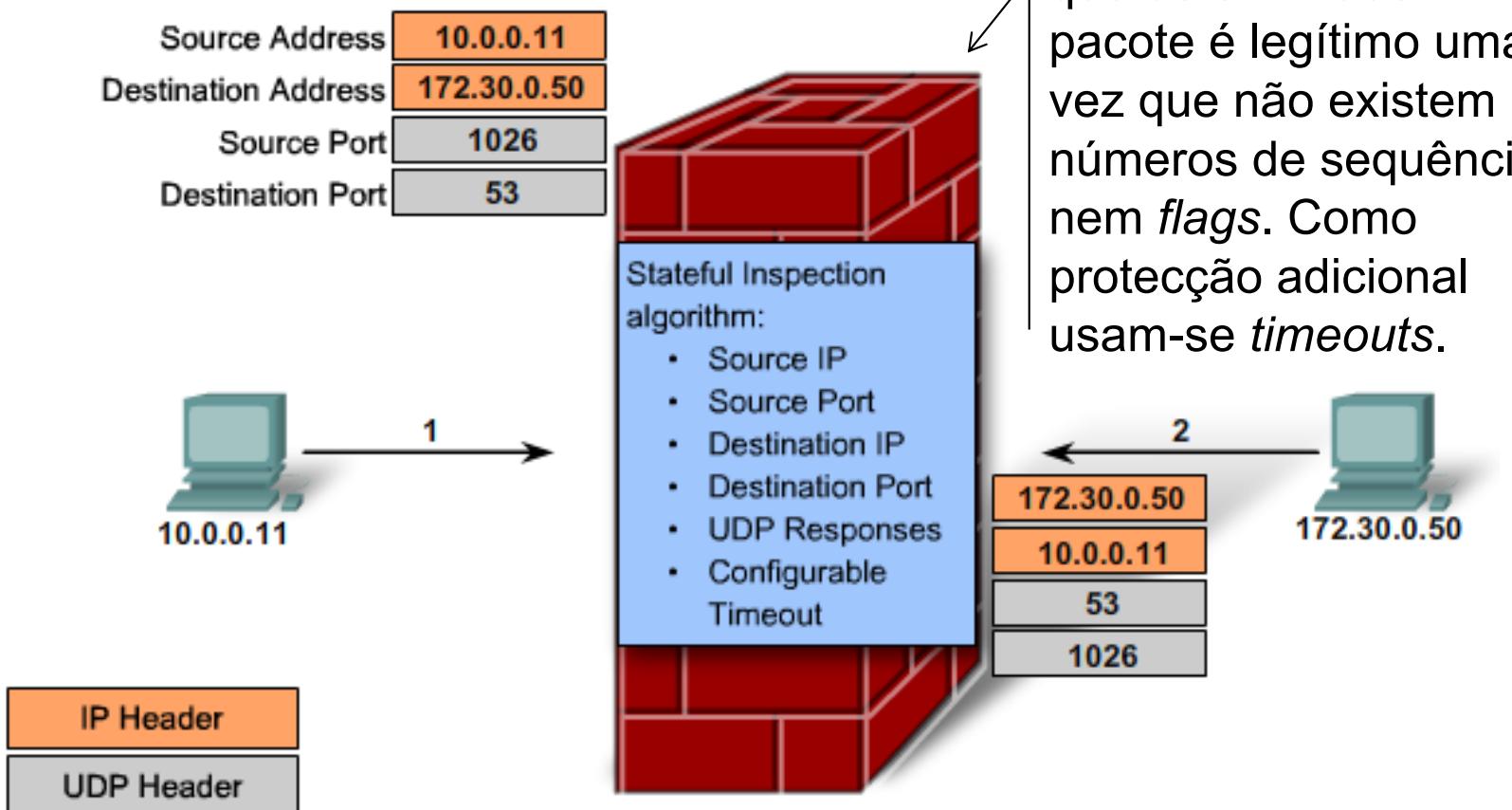
CBAC - Operação

- Vigilância genérica de sessões TCP



CBAC - Operação

- Vigilância genérica de sessões UDP



CBAC - Operação

- Serviços prestados
 - Traffic filtering
 - A filtragem de tráfego TCP/UDP é feita com base em informação de sessão extraída da camada de aplicação
 - Traffic inspection
 - A evolução das sessões é acompanhada para detetar situações potencialmente perigosas (ex. ataques DoS).
 - Alerts and Audit Trails
 - É possível programar o IOS para gerar alertas e relatórios de auditoria específicos para determinados tipos de tráfego.
 - Intrusion Prevention
 - A prevenção de intrusões do CBAC encontra-se limitada a alguns padrões de ataques comuns sobre tráfego SMTP.

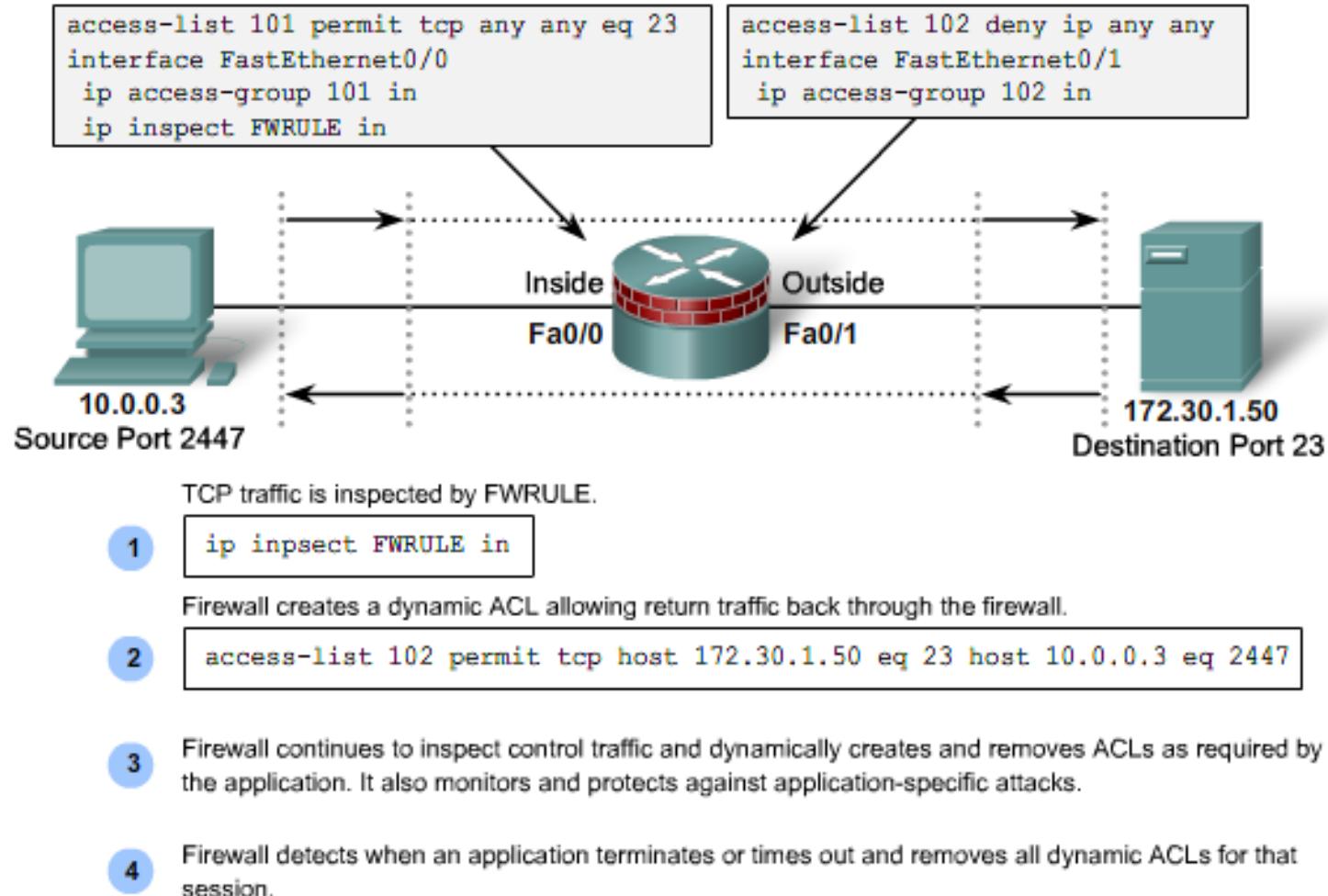


CBAC - Operação

- Outros protocolos IP
 - Apenas são tratados protocolos baseados no TCP e UDP. Outro tipo de tráfego IP (ex. ICMP) não é coberto!
 - As *firewalls* baseadas no estado não suportam tráfego em trânsito encapsulado em túneis GRE ou IPSec. No entanto se o router for endpoint do túnel o tráfego interno é convenientemente tratado.
 - Quando é dado algum suporte a este tipo de protocolos a estratégia é semelhante à que o CBAC emprega no UDP
 - Em protocolos complexos as *firewalls* com estado interpretam o conteúdo de nível aplicacional que viaja nos pacotes iniciais da sessão (*snooping*) de modo a inferir que sessões adicionais são esperadas no seu seguimento.
 - Esta técnica é conhecida por *Deep Packet Inspection (DPI)*



CBAC - Operação



CBAC - Operação

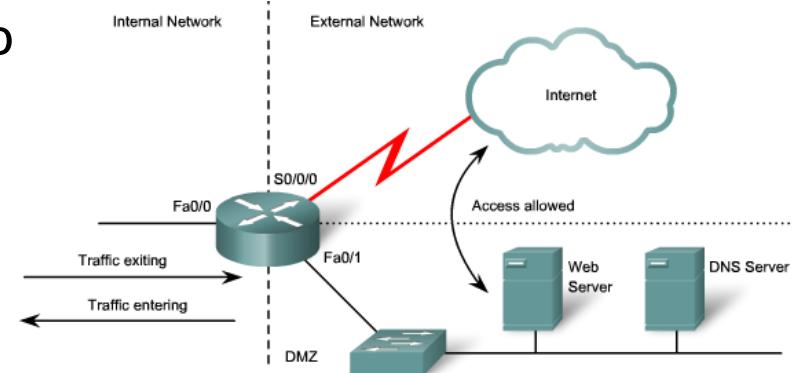
- Protecção de ataques de nível aplicacional
 - São aceites apenas primitivas protocolares legítimas (comandos SMTP)
- Quando um ataque é detectado, o *router* pode
 - Gerar mensagens de alerta
 - Proteger o bom desempenho do *router*
 - Bloquear pacotes suspeitos
- O estado das sessões é gerido com base em *timeouts* e limiares (*thresholds*) para se proteger contra ataques DoS
 - Número total de sessões em abertura (*half-opened*)
 - Número de sessões em abertura por intervalo de tempo
 - Número de sessões em abertura por endereço IP
 - Em resposta a uma ultrapassagem do limiar o *router* envia *resets*, fechando as sessões problemáticas mais antigas, e bloqueia temporariamente o estabelecimento de novas sessões



CBAC - Configuração

1. Selecção de interfaces para iniciar as actividades de inspecção

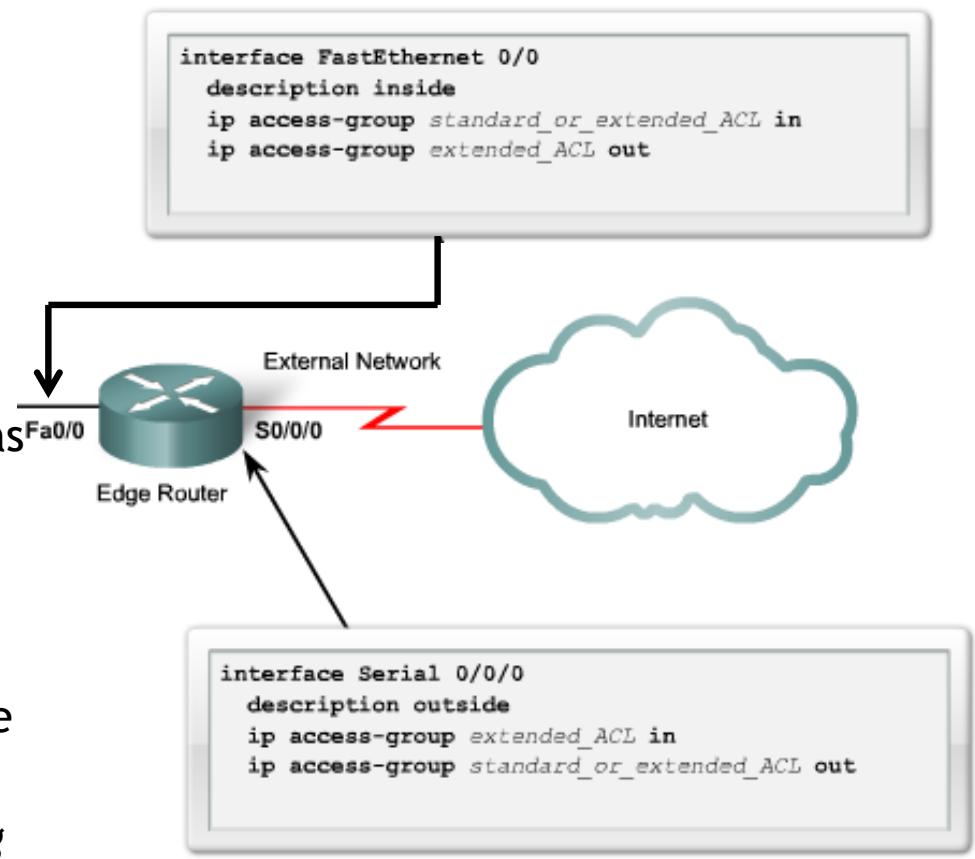
- Identificar a interface interna
 - Para o CBAC é a interface por onde pode ser iniciada uma sessão
- Identificar a interface externa
 - Sessões iniciadas numa interface externa são bloqueadas
- Por vezes pode ser necessário configurar o CBAC em ambos os sentidos (e.g., comunicação com a intranet/extranet)
 - Neste caso é aconselhável primeiro tratar um sentido e depois outro, classificando as interfaces como internas e externas, sucessivamente



CBAC - Configuração

2. Configurar ACLs em todas as interfaces

- Partir de uma configuração base simples: tráfego pode fluir da rede protegida para a não protegida, mas não em sentido oposto
- Permitir tráfego que deve ser sujeito a inspecção pelo CBAC
- Aplicar ACLs *extended* para filtrar tráfego que venha de redes não protegidas de modo que as entradas temporárias geradas pelo CBAC possam ser criadas
 - Em routers com duas interfaces usar ACLs *inbounds* em ambas.
- Configurar entradas *anti-spoofing* e *anti-smurf*
- Explicitar o **deny** implícito com *log*



CBAC - Configuração

3. Definir as regras de inspecção

- Normalmente é necessário definir uma única regra de inspecção.
- Quando é preciso explorar os dois sentidos são necessárias duas.

Router(config)#

```
ip inspect name inspection_name protocol [alert {on | off}] [audit-trail {on | off}]
[timeout seconds]
```

Parameter	Description
<code>inspection-name</code>	Names the set of inspection rules. If you want to add a protocol to an existing set of rules, use the same inspection name for the rules.
<code>protocol</code>	The protocol to inspect.
<code>alert {on off}</code>	(Optional) For each inspected protocol, the generation of alert messages can be set to on or off. If no option is selected, alerts are generated based on the setting of the <code>ip inspect alert-off</code> command.
<code>audit-trail {on off}</code>	(Optional) For each inspected protocol, the <code>audit-trail</code> option can be set to on or off. If no option is selected, <code>audit trail</code> messages are generated based on the setting of the <code>ip inspect audit-trail</code> command.
<code>timeout seconds</code>	(Optional) Specify the number of seconds for a different idle timeout to override the global TCP or UDP idle timeouts for the specified protocol. This timeout overrides the global TCP and UDP timeouts but does not override the global Domain Name Service (DNS) timeout.



CBAC - Configuração

3. Definir as regras de inspecção: exemplos

- Inspecção de tráfego SMTP e FTP com *timeout* de 300 segundos

```
Router (config)# ip inspect name FWRULE smtp alert on audit-trail on  
timeout 300  
Router (config)# ip inspect name FWRULE ftp alert on audit-trail on  
timeout 300
```

- Inspecção de tráfego HTTP e autorização para download de *applets*

```
Router (config)# ip inspect name PERMIT_JAVA http java-list 10  
Router (config)# access-list 10 permit 10.224.10.0 0.0.0.255
```

- Inspecção de um conjunto alargado de protocolos e definido um *timeout* alargado (12 horas) para o TCP

```
Router (config)# ip inspect name in2out rcmd  
Router (config)# ip inspect name in2out ftp  
Router (config)# ip inspect name in2out tftp  
Router (config)# ip inspect name in2out tcp timeout 43200  
Router (config)# ip inspect name in2out http  
Router (config)# ip inspect name in2out udp
```



CBAC - Configuração

4. Aplicação das regras de inspecção

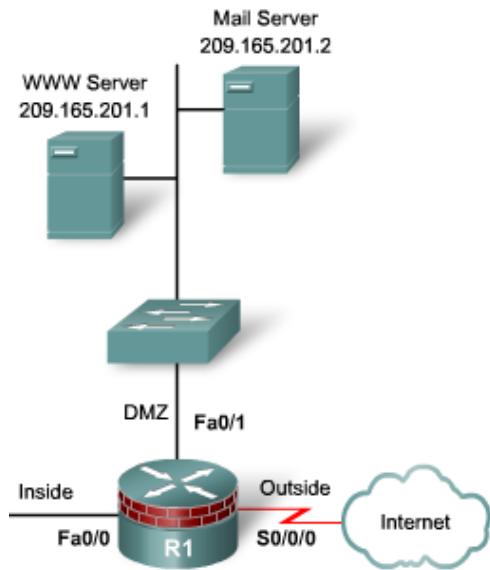
- Número total de sessões TCP/UDP *half-open*
- Número de sessões *half-open* com base em tempo
- Número de sessões *half-open* TCP
- Se *half opens* existente > **max-incompletehigh**
 - IOS apaga sessões existentes para acomodar novas sessões até chegar ao limiar **max-incompletelow**
- Se ritmo de novas sessões > **one-minutehigh** então
 - IOS apaga sessões até que o ritmo de sessões novas seja inferior a **one-minutelow**



CBAC - Configuração

4. Exemplo

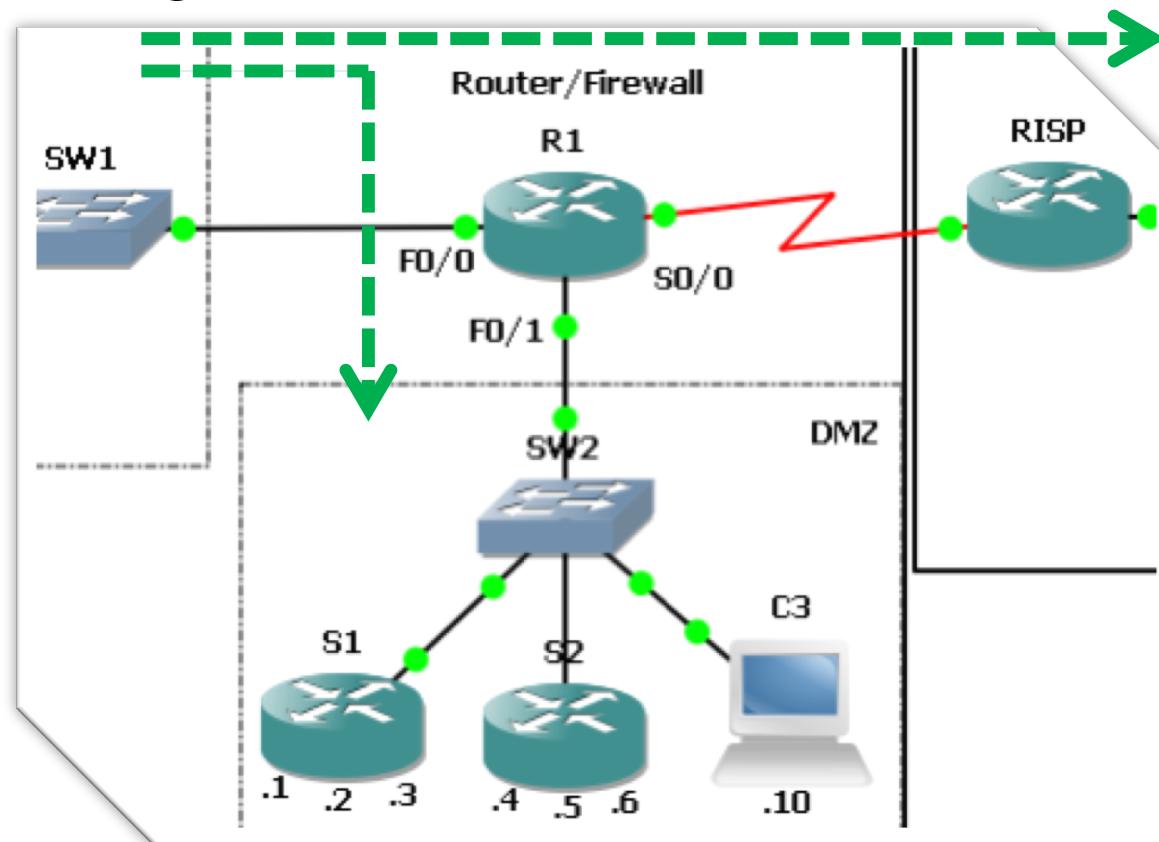
```
ip inspect name MYSITE tcp
ip inspect name MYSITE udp
!
interface FastEthernet0/0
    ip address 10.10.10.254 255.255.255.0
    ip access-group 101 in
    ip inspect MYSITE in
!
interface FastEthernet0/1
    ip address 209.165.201.30 255.255.255.224
!
interface Serial0/0/0
    ip address 209.165.200.225 255.255.255.224
    ip access-group 102 in
!
access-list 101
    permit tcp 10.10.10.0 0.0.0.255 any
    permit udp 10.10.10.0 0.0.0.255 any
    permit icmp 10.10.10.0 0.0.0.255 any
    deny ip any any
!
access-list 102
```



```
permit tcp any host 209.165.201.1 eq www
permit tcp any host 209.165.201.2 eq smtp
permit icmp any any echo-reply
permit icmp any any unreachable
permit icmp any any administratively-
    prohibited
permit icmp any any packet-too-big
permit icmp any any echo
permit icmp any any time-exceeded
deny ip any any
```

CBAC - Desafio

- Programe R1 de modo que apenas possa entrar na intranet tráfego UDP/TCP/ICMP de sessões aí iniciadas



CBAC - Solução

1. Programar as ACLs devidamente

```
R1-Firewall(config)#ip access-list extended INTRANET-DMZ_INTERNET
R1-Firewall(...)#permit tcp 192.168.1.0 0.0.0.255 194.65.52.0 0.0.0.15
R1-Firewall(...)#permit udp 192.168.1.0 0.0.0.255 194.65.52.0 0.0.0.15
R1-Firewall(...)#permit icmp 192.168.1.0 0.0.0.255 194.65.52.0 0.0.0.15
R1-Firewall(config-ext-nacl)#permit tcp 194.65.52.128 0.0.0.127 any
R1-Firewall(config-ext-nacl)#permit udp 194.65.52.128 0.0.0.127 any
R1-Firewall(config-ext-nacl)#permit icmp 194.65.52.128 0.0.0.127 any
R1-Firewall(config-ext-nacl)#exit
R1-Firewall(config)#ip access-list extended DMZ-INTRANET_INTERNET
R1-Firewall(config-ext-nacl)#deny ip any any
R1-Firewall(...)#ip access-list extended INTERNET-DMZ_INTRANET
R1-Firewall(config-ext-nacl)#deny ip any any
R1-Firewall(config-ext-nacl)#exit
R1-Firewall(config)#interface f0/0
R1-Firewall(config-if)#ip access-group INTRANET-DMZ_INTERNET in
R1-Firewall(config-if)#interface f0/1
R1-Firewall(config-if)#ip access-group DMZ-INTRANET_INTERNET in
R1-Firewall(config-if)#interface s0/0
R1-Firewall(config-if)#ip access-group INTERNET-DMZ_INTRANET in
```

Esta ACL define o tráfego sujeito a inspecção pela *firewall stateful*.

Estas ACL vedam as interfaces a sessões não inspecionadas

CBAC - Solução

2. Activar depuração na *firewall*

```
R1-Firewall#debug ip packet  
IP packet debugging is on
```

3. Verificar que a comunicação entre as duas redes IP da intranet se encontra vedada

```
VPCS[1]> ping 194.65.52.129  
194.65.52.129 icmp_seq=1 timeout  
194.65.52.129 icmp_seq=2 timeout  
194.65.52.129 icmp_seq=3 timeout  
194.65.52.129 icmp_seq=4 timeout  
194.65.52.129 icmp_seq=5 timeout
```

```
R1-Firewall#  
*Mar 1 00:29:34.943: IP: s=192.168.1.1 (FastEthernet0/0), d=194.65.52.129, len 48,  
access denied  
*Mar 1 00:29:34.947: IP: tableid=0, s=194.65.52.254 (local), d=192.168.1.1  
(FastEthernet0/0), routed via FIB  
*Mar 1 00:29:34.951: IP: s=194.65.52.254 (local), d=192.168.1.1 (FastEthernet0/0),  
len 56, sending
```



CBAC - Solução

3. (cont.)

- Neste caso é a ACL de entrada de R1.F0/0 que impede de imediato o tráfego entre as duas redes da intranet.

4. Verificar que, para já, o tráfego de retorno da comunicação da intranet para a DMZ se encontra vedado

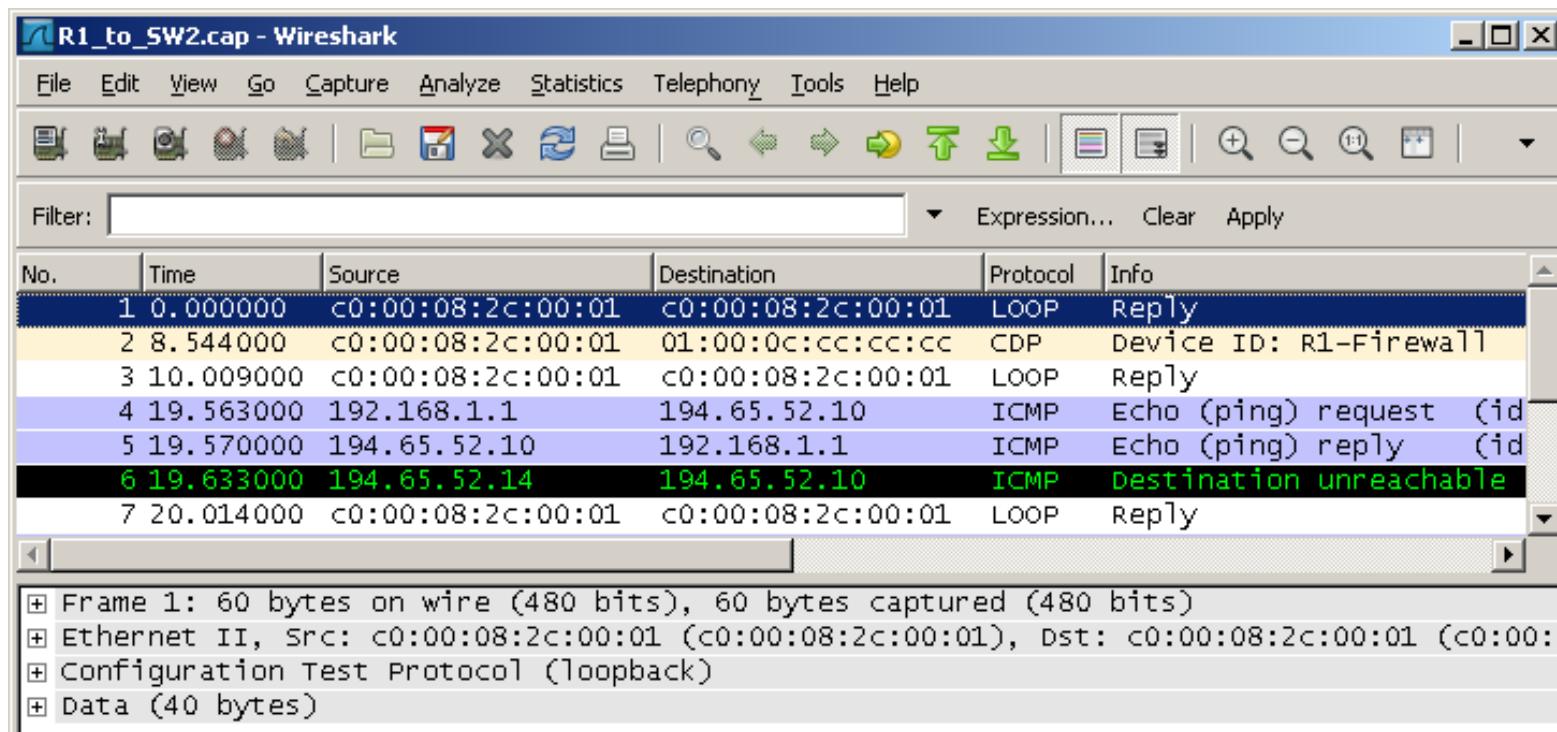
```
VPCS[1]> ping 194.65.52.10
194.65.52.10 icmp_seq=1 timeout
194.65.52.10 icmp_seq=2 timeout
194.65.52.10 icmp_seq=3 timeout
194.65.52.10 icmp_seq=4 timeout
194.65.52.10 icmp_seq=5 timeout
```

```
R1-Firewall#
*Mar 1 00:17:57.483: IP: s=194.65.52.10 (FastEthernet0/1), d=192.168.1.1, len 48,
access denied
*Mar 1 00:17:57.487: IP: tableid=0, s=194.65.52.14 (local), d=194.65.52.10
(FastEthernet0/1), routed via FIB
*Mar 1 00:17:57.491: IP: s=194.65.52.14 (local), d=194.65.52.10 (FastEthernet0/1),
len 56, sending
```



CBAC - Solução

4. (cont.) Torna-se mais claro, através de uma captura de tráfego sobre a interface R1.F0/1, que o *router* aceita encaminhar num sentido mas não no inverso



CBAC - Solução

5. Da mesma forma encontra-se vedado o trânsito de tráfego de retorno de sessões estabelecidas entre a intranet e a Internet

```
VPCS[2]> ping 4.4.4.4
4.4.4.4 icmp_seq=1 timeout
4.4.4.4 icmp_seq=2 timeout
4.4.4.4 icmp_seq=3 timeout
4.4.4.4 icmp_seq=4 timeout
4.4.4.4 icmp_seq=5 timeout
```

```
R1-Firewall#
*Mar 1 00:34:24.187: IP: s=4.4.4.4 (Serial0/0), d=194.65.52.129, len 48, access denied
*Mar 1 00:34:24.191: IP: tableid=0, s=194.65.53.1 (local), d=4.4.4.4 (Serial0/0),
routed via FIB
*Mar 1 00:34:24.195: IP: s=194.65.53.1 (local), d=4.4.4.4 (Serial0/0), len 56,
sending
```



CBAC - Solução

5. Vamos tornar o comportamento da *firewall statefull* (i.e., activar a inspecção de sessões TCP, UDP, ICMP que o router aceite receber pela sua interface F0/0)

```
R1-Firewall(config)#ip inspect name FROM-INTRANET tcp  
R1-Firewall(config)#ip inspect name FROM-INTRANET udp  
R1-Firewall(config)#ip inspect name FROM-INTRANET icmp  
R1-Firewall(config)#interface f0/0  
R1-Firewall(config-if)#ip inspect FROM-INTRANET in →  
R1-Firewall(config-if)#^z
```

Só chegará a ser inspeccionado o tráfego que a ACL *in* desta interface permitir.

6. Consultar o estado da *firewall statefull*

```
R1-Firewall#show ip inspect ?  
all          Inspection all available information  
config       Inspection configuration  
interfaces   Inspection interfaces  
mib          FW MIB specific show commands  
name         Inspection name  
sessions     Inspection sessions  
statistics   Inspection statistics
```



CBAC - Solução

6. (cont.)

```
R1-Firewall#show ip inspect all
Session audit trail is disabled
Session alert is enabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
  Inspection name FROM-INTRANET
    tcp alert is on audit-trail is off timeout 3600
    udp alert is on audit-trail is off timeout 30
    icmp alert is on audit-trail is off timeout 10
Interface Configuration
  Interface FastEthernet0/0
    Inbound inspection rule is FROM-INTRANET
      tcp alert is on audit-trail is off timeout 3600
      udp alert is on audit-trail is off timeout 30
      icmp alert is on audit-trail is off timeout 10
    Outgoing inspection rule is not set
    Inbound access list is INTRANET-DMZ_INTERNET
    Outgoing access list is not set
```

Prevenção de ataques DoS

Alerts, audit-trails e prazos de indolência das sessões (seg.)

CBAC - Solução

7. Verificar agora o funcionamento de sessões iniciadas na intranet destinadas à DMZ

```
VPCS[1]> ping 194.65.52.10
194.65.52.10 icmp_seq=1 ttl=63 time=99.000 ms
194.65.52.10 icmp_seq=2 ttl=63 time=50.000 ms
194.65.52.10 icmp_seq=3 ttl=63 time=27.000 ms
194.65.52.10 icmp_seq=4 ttl=63 time=37.000 ms
194.65.52.10 icmp_seq=5 ttl=63 time=28.000 ms
```

```
R1-Firewall#
*Mar 1 00:59:06.203: IP: tableid=0, s=192.168.1.1 (FastEthernet0/0),
d=194.65.52.10 (FastEthernet0/1), routed via FIB
*Mar 1 00:59:06.211: IP: s=192.168.1.1 (FastEthernet0/0), d=194.65.52.10
(FastEthernet0/1), g=194.65.52.10, len 48, forward
```

```
R1-Firewall#show ip inspect sessions
```

Established Sessions

Session 64850E50 (192.168.1.1:8)=>(194.65.52.10:0) icmp SIS OPEN

```
R1-Firewall#show ip inspect sessions
```

←
Dez segundos depois
expira a sessão ICMP.

CBAC - Solução

8. Verificar o comportamento perante sessões TCP iniciadas na intranet para DMZ

```
VPCS[2]> ping 194.65.52.10 -3
Connect 7@194.65.52.10 seq=1 ttl=63 time=73.000 ms
SendData 7@194.65.52.10 timeout
Connect 7@194.65.52.10 timeout
Connect 7@194.65.52.10 timeout
Connect 7@194.65.52.10 timeout
```

As sessões TCP iniciadas pelo VPCS são consideradas “estranhas” pela firewall progredindo apenas até à fase de estabelecimento da sessão TCP

```
R1-Firewall#show ip inspect sessions detail
Half-open Sessions
Session 64850E50 (194.65.52.129:7)=>(194.65.52.10:7) tcp SIS_OPENING
Created 00:00:05, Last heard 00:00:00
Bytes sent (initiator:responder) [0:0]
In SID 194.65.52.10[7:7]=>194.65.52.129[7:7] on ACL DMZ-INTRANET__INTERNET
(3 matches)
```



Identificação da ACL “aberta” dinamicamente para que o tráfego de retorno pudesse passar.

CBAC - Solução

9. Verificar o comportamento perante sessões UDP iniciadas na intranet para DMZ

```
VPCS[2]> ping 194.65.52.10 -2
194.65.52.10 udp_seq=1 ttl=63 time=46.000 ms
194.65.52.10 udp_seq=2 ttl=63 time=47.000 ms
194.65.52.10 udp_seq=3 ttl=63 time=26.000 ms
194.65.52.10 udp_seq=4 ttl=63 time=23.000 ms
194.65.52.10 udp_seq=5 ttl=63 time=30.000 ms
```

```
R1-Firewall#show ip inspect sessions details
Established Sessions
Session 64850E50 (194.65.52.129:7)=>(194.65.52.10:7) udp SIS_OPEN
Created 00:00:06, Last heard 00:00:02
Bytes sent (initiator:responder) [100:100]
In SID 194.65.52.10[7:7]=>194.65.52.129[7:7] on ACL DMZ-INTRANET__INTERNET
(5 matches)
```

CBAC - Troubleshooting

- Existem dois mecanismos de *feedback* relativamente à operação das CBAC:
 - Alertas (*alerts*)
 - São usados pelo IOS para anunciar condições críticas como nível baixo de recursos do *router*, ataques DoS em curso, PDUs suspeitas, etc.

```
02:04:55: %FW-4-TCP_MAJORDOMO_EXEC_BUG: Sig:3107:  
Majordomo Execute Attack - from 209.165.201.5 to 192.168.1.1:
```
 - Encontram-se activos por omissão e são reportados na consola.
 - Nota: Se tivermos a monitorar o router remotamente, através uma sessão SSH/Telnet, podemos solicitar igualmente a entrega de mensagens que por omissão apenas são anunciadas na consola

```
Router (config)# terminal monitor
```
 - Embora não seja recomendável, os alertas podem ser desligados globalmente e activados selectivamente, por regra de inspecção.

```
Router (config)# ip inspect alert-off
```



CBAC - Troubleshooting

- (cont.) Existem dois mecanismos de feedback relativamente à operação das CBAC:

- Auditorias (*audits*)

- Fornecem informação relativamente às sessões que a *firewall* inspecciona e acompanha (aceitação de sessão, rejeição de sessão, ...)
 - Por omissão a informação é reportada na consola mas pode ser dirigida a um servidor de *logs*.
 - Este mecanismos informativo encontra-se inactivo por omissão mas pode ser activado através do comando:

```
Router (config)#ip inspect audit-trail
```

```
R1(config)# logging on
R1(config)# logging host 10.0.0.3
R1(config)# ip inspect audit-trail
R1(config)# no ip inspect alert-off
```

```
*Mar 1 02:40:14.499: %FW-6-SESS_AUDIT_TRAIL_START: Start udp session:
initiator (192.168.1.1:7) -- responder (194.65.52.10:7)
```



CBAC - Troubleshooting

- Depuração detalhada do CBAC

```
Router#debug ip inspect parameter
```

Parameter	Explanation
<code>tcp</code>	Displays TCP inspection events.
<code>udp</code>	Displays UDP inspection events.
<code>icmp</code>	Displays ICMP inspection events.
<code>application_name</code>	Displays inspection events for the specified application, such as TFTP or SMTP.
<code>events</code>	Displays CBAC events, including the processing of packets.
<code>object-creation</code>	Displays information about an entry being added to the state table.
<code>object-deletion</code>	Displays information about an entry being removed to the state table.
<code>function-trace</code>	Displays information about the software functions that CBAC calls.
<code>timers</code>	Displays information related to CBAC timers, such as information that the TCP or UDP idle timers are reached.
<code>detailed</code>	Displays information about all the CBAC processes on the router.

aol-msgr, cuseeme, dns, ftp-cmd, ftp-token, h323, http, icmp, imap, msn-msgr, netshow, pop3, rcmd, realaudio, rpc, rtsp, sip, skinny, smtp, sqlnet, streamworks, tcp, tftp, timers, udp, vdolive, yahoo-msgr



```
Router#debug policy-firewall (IOS ≥ 12.4(20)T)
```



CBAC - Troubleshooting

- (cont.) O que estará errado com as sessões TCP do VPCS?

```
R1-Firewall#debug ip inspect tcp
INSPECT TCP Inspection debugging is on
R1-Firewall#
*Mar  1 02:59:17.711: %FW-6-SESS_AUDIT_TRAIL_START: Start tcp session:
initiator (192.168.1.1:7) -- responder (194.65.52.10:7)
*Mar  1 02:59:17.715: CBAC* sis 64850B98 pak 63D3AE3C SIS_CLOSED/LISTEN TCP
SYN SEQ 1307532979 LEN 0 (192.168.1.1:7) => (194.65.52.10:7)
*Mar  1 02:59:17.727: CBAC* sis 64850B98 pak 63E64950 SIS_OPENING/SYNSENT
TCP SYN ACK 1307532980 SEQ 1540383426 LEN 0 (194.65.52.10:7) <=
(192.168.1.1:7)
*Mar  1 02:59:17.775: CBAC* sis 64850B98 pak 63D3AE3C SIS_OPENING/SYNRCVD
TCP SYN ACK 1540383427 SEQ 1307532979 LEN 0 (192.168.1.1:7) =>
(194.65.52.10:7)
*Mar  1 02:59:17.779: CBAC* sis 64850B98 L4 inspect result: DROP packet
63D3AE3C (192.168.1.1:7) (194.65.52.10:7) bytes 0 ErrStr = Invalid Segment
tcp
```



CBAC - Troubleshooting

- (cont.) O que estará errado com as sessões TCP do VPCS?

R1_to_SW2.cap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: ip.proto == 6

No. Time Source Destination Protocol Info

No.	Time	Source	Destination	Protocol	Info
4	17.337000	192.168.1.1	194.65.52.10	TCP	7 > 7 [SYN] Seq=1307533166 Win=65535 Len=0
5	17.347000	194.65.52.10	192.168.1.1	TCP	7 > 7 [SYN, ACK] Seq=294702567 Ack=1307533167 Win=65535 Len=0
6	19.619000	192.168.1.1	194.65.52.10	TCP	7 > 7 [SYN] Seq=1307533166 Win=65535 Len=0
7	19.628000	194.65.52.10	192.168.1.1	TCP	7 > 7 [SYN, ACK] Seq=1726956429 Ack=1307533167 Win=65535 Len=0
9	21.837000	192.168.1.1	194.65.52.10	TCP	7 > 7 [SYN] Seq=1307533166 Win=65535 Len=0
10	21.845000	194.65.52.10	192.168.1.1	TCP	7 > 7 [SYN, ACK] Seq=336465782 Ack=1307533167 Win=65535 Len=0
11	24.041000	192.168.1.1	194.65.52.10	TCP	7 > 7 [SYN] Seq=1307533166 Win=65535 Len=0
12	24.052000	194.65.52.10	192.168.1.1	TCP	7 > 7 [SYN, ACK] Seq=861021530 Ack=1307533167 Win=65535 Len=0
13	26.241000	192.168.1.1	194.65.52.10	TCP	7 > 7 [SYN] Seq=1307533166 Win=65535 Len=0
14	26.251000	194.65.52.10	192.168.1.1	TCP	7 > 7 [SYN, ACK] Seq=278722862 Ack=1307533167 Win=65535 Len=0

Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: c0:00:08:2c:00:01 (c0:00:08:2c:00:01), Dst: 00:50:79:66:68:02 (00:50:79:66:68:02)
Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 194.65.52.10 (194.65.52.10)
Transmission Control Protocol, Src Port: 7 (7), Dst Port: 7 (7), Seq: 1307533166, Len: 0
Source port: 7 (7)
Destination port: 7 (7)
[Stream index: 0]
Sequence number: 1307533166
Acknowledgement number: Broken TCP. The acknowledge field is nonzero while the ACK flag is not set
Header length: 40 bytes
Flags: 0x02 (SYN)
Window size: 65535
Checksum: 0x0000 0x14d4+4e00 d4-5b7add

CBAC - Afinações de *timeouts* e *thresholds*

Timeout or Threshold Value to Change	Command	Default
The length of time the software waits for a TCP session to reach the established state before dropping the session.	<code>ip inspect tcp synwait-time <i>seconds</i></code>	30 seconds
Disable the window scale option check for a TCP packet that has an invalid window scale option under the Context-Based Access Control (CBAC) firewall.	<code>ip inspect tcp window-scale-enforcement loose</code>	The strict window scale option check is enabled in the firewall by default.
The length of time a TCP session will still be managed after the firewall detects a FIN-exchange.	<code>ip inspect tcp finwait-time <i>seconds</i></code>	5 seconds
The length of time a TCP session will still be managed after no activity (the TCP idle timeout). ¹	<code>ip inspect tcp idle-time <i>seconds</i></code>	3600 seconds (1 hour)



CBAC - Afinações de *timeouts* e *thresholds*

The length of time a UDP session will still be managed after no activity (the UDP idle timeout). ¹	idle-time <i>ip inspect udp seconds</i>	30 seconds
The length of time a DNS name lookup session will still be managed after no activity.	dns-timeout <i>seconds</i> ip inspect	5 seconds
The number of existing half-open sessions that will cause the software to start deleting half-open sessions. ²	max-incomplete <i>high number</i> ip inspect	500 existing half-open sessions
		400 existing half-open sessions



CBAC - Afinações de *timeouts* e *thresholds*

Timeout or Threshold Value to Change	Command	Default
The number of existing half-open sessions that will cause the software to stop deleting half-open sessions. 2	ip inspect max-incomplete low <i>number</i>	
The rate of new sessions that will cause the software to start deleting half-open sessions. 2	ip inspect one-minute high <i>number</i>	500 half-open sessions per minute
The rate of new sessions that will cause the software to stop deleting half-open sessions. 2	ip inspect one-minute low <i>number</i>	400 half-open sessions per minute



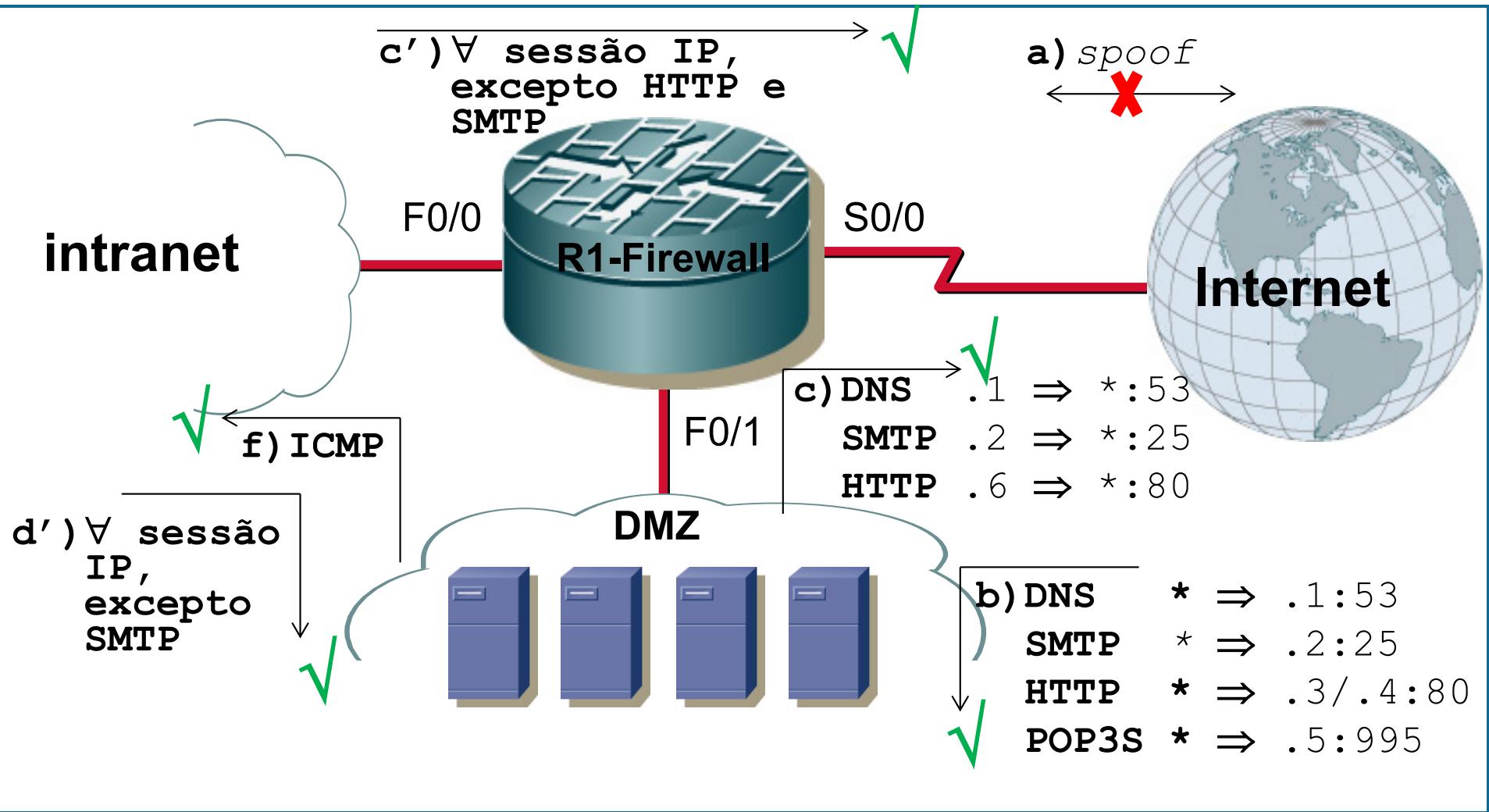
CBAC - Afinações de *timeouts* e *thresholds*

The number of existing half-open TCP sessions with the same destination host address that will cause the software to start dropping half-open sessions to the same destination host address. ³	ip inspect tcp max-incomplete host <i>number</i> block-time <i>minutes</i>	50 existing half-open TCP sessions; 0 minutes
---	---	---



CBAC - Desafio

(satisfça as políticas de segurança atrás enunciadas com uma *firewall* CBAC)



CBAC - Solução

- TPC



Zone-based Policy Firewall (ZPF ≡ ZBF ≡ ZFW)

DEIS

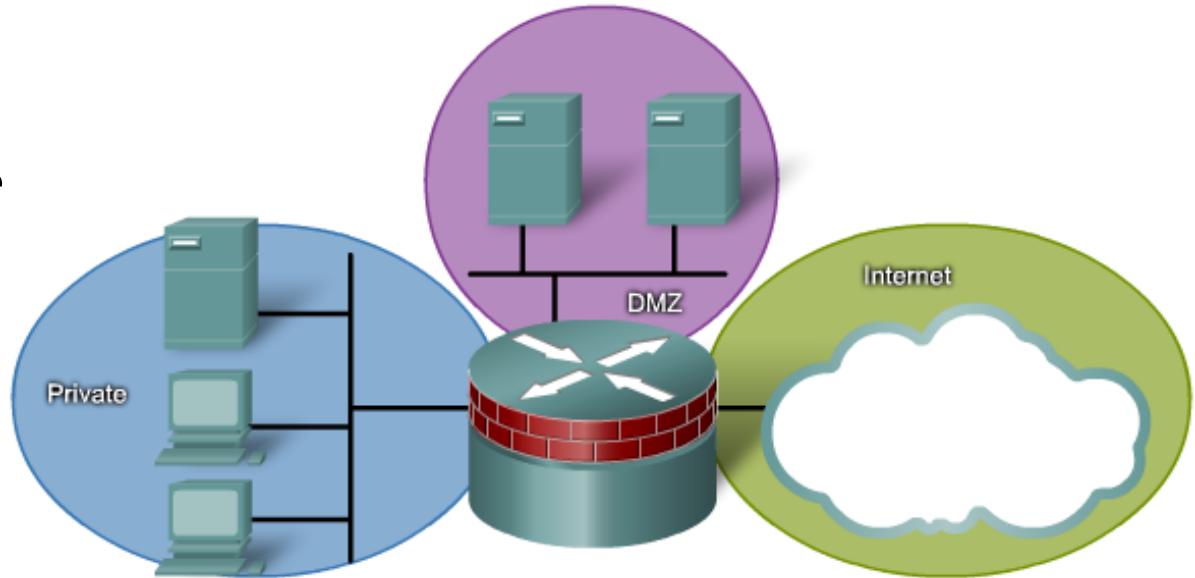
ZPF: Introdução

- Modelo de *firewall* introduzido pela Cisco em 2006 (IOS \geq 12.4(6)T)
- A visão de rede é consideravelmente simplificada:
 - As interfaces passam a ser agrupadas por zonas
 - As políticas de inspecção são aplicadas a fluxos caracterizados pela zona origem (X) e destino (Y)
 - Existe, por omissão, uma proibição implícita de fluxo entre qualquer par de zonas
 - Estratégia semelhante ao “deny” implícito no final das ACL.
- As funcionalidades do CBAC continuam a ser suportadas
 - Inspecção *stateful* de pacotes, análise profunda de pacotes (camada de aplicação), filtragem de URLs, mitigação de DoS



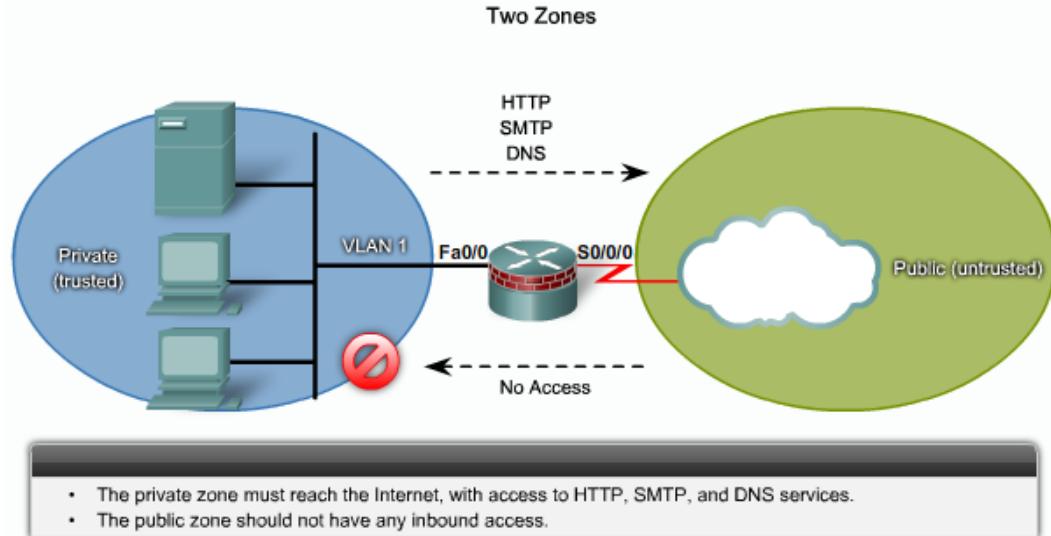
ZPF: Introdução

- Orientação à Zona
 - Se fosse adicionada uma segunda interface à zona Private então todos os terminais servidos por ela poderiam comunicar livremente com os terminais servidos pela primeira interface.
 - Adicionalmente, os novos terminais da zona *Private* passavam de imediato a comungar das mesmas políticas de controlo de tráfego a que o primeiro grupo estava sujeito
- As políticas passam a ser expressas na ***Cisco Common Classification Policy Language (C3PL)***, uma linguagem com estrutura hierárquica.



ZPF: Introdução

- Orientação à Zona
 - O seu funcionamento não está ligado às ACLs
 - Por omissão é negada toda a comunicação entre zonas distintas
 - As políticas tornam-se mais legíveis (C3PL)



- Principais desvantagens do CBAC

- Múltiplas políticas de inspecção de tráfego dispersas por múltiplas interfaces aumentam de forma dramática a dificuldade de interpretação
- As políticas não podem ser restringidas a um grupo de terminais ou subrede uma vez que todo o tráfego que atravesse uma interface é sujeito à mesma inspecção
- O processo assenta demasiado nas ACLs



ZPF: Abordagem

1. Identificar as Zonas

- No particionamento em zonas apenas são tidas em consideração os diversos níveis de segurança a considerar na nossa rede

2. Estabelecer as políticas de segurança inter-zonal

- Para cada par de zonas {origem→destino} identificar que sessões podem os clientes na zona de origem estabelecer com os servidores da zona destino
- Para tráfego que não siga o conceito de sessão (e.g., *IPSec Encapsulating Security Payload (ESP)*) o administrador deve definir fluxos unidireccionais da fonte para a origem e vice-versa



ZPF: Abordagem

3. Projectar a infra-estrutura física

- Partindo das considerações de segurança reunidas nos passos anteriores e tendo em conta aspectos como a disponibilidade o administrador deve projectar a infra-estrutura física das *firewalls*

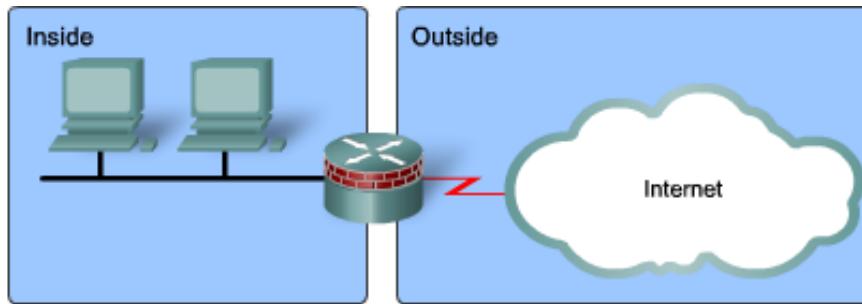
4. Identificar subconjuntos dentro de zonas e fundir os requisitos de tráfego

- Para cada *firewall* prevista no projecto devem ser identificados subconjuntos de zonas servidos pelas suas interfaces.
- Por exemplo, pode suceder que múltiplas zonas sejam indirectamente servidas pela mesma interface de uma *firewall*, o que resulta numa política inter-zonal específica do dispositivo

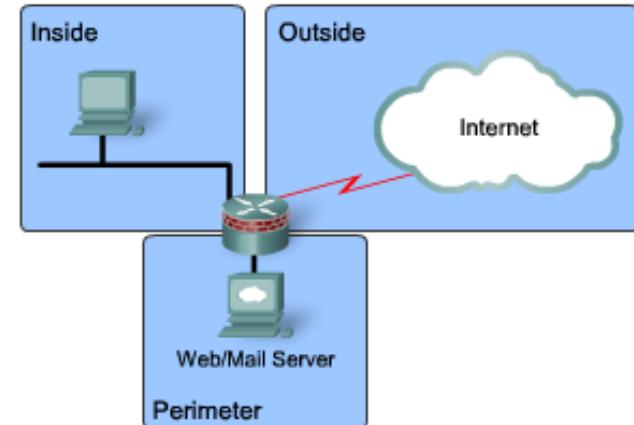


ZPF: Arquitecturas típicas

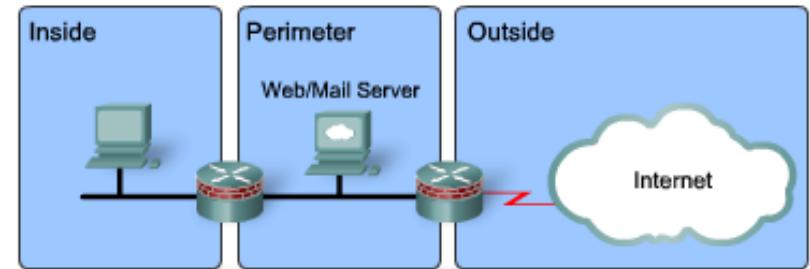
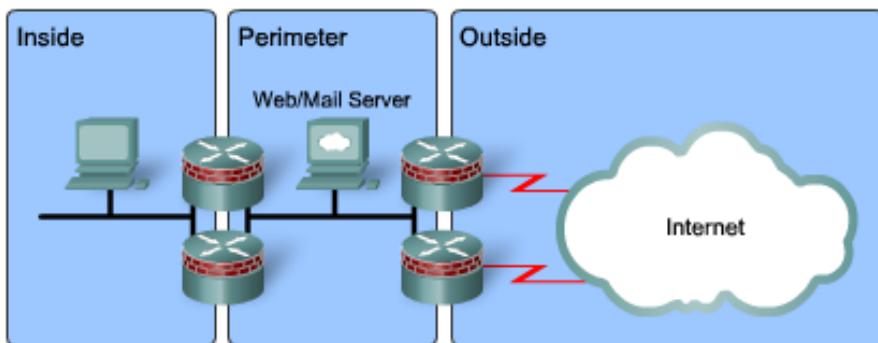
- Lan-Internet



- Servidores públicos

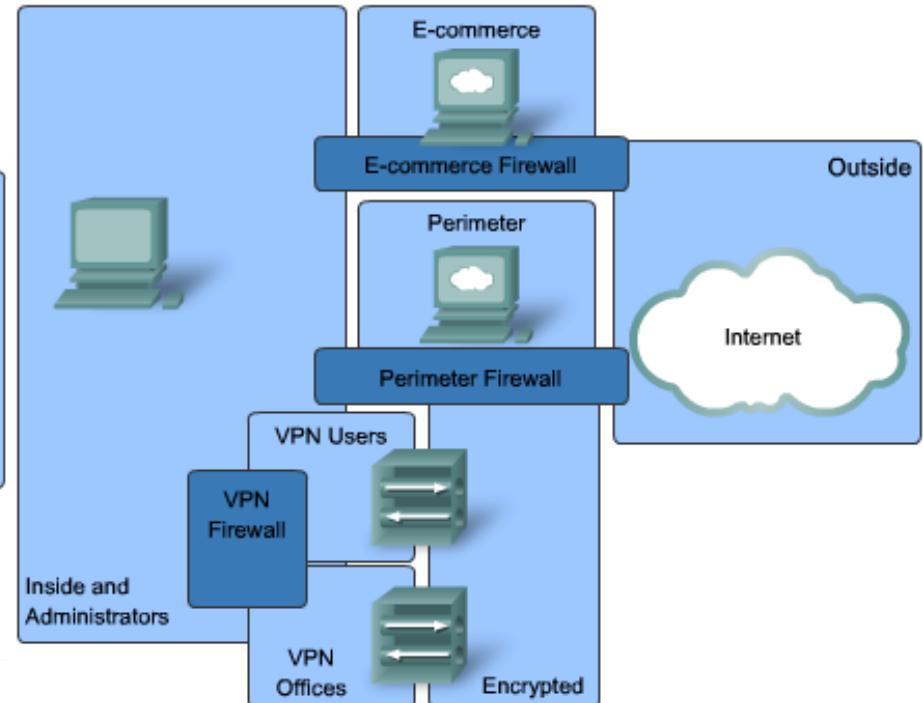
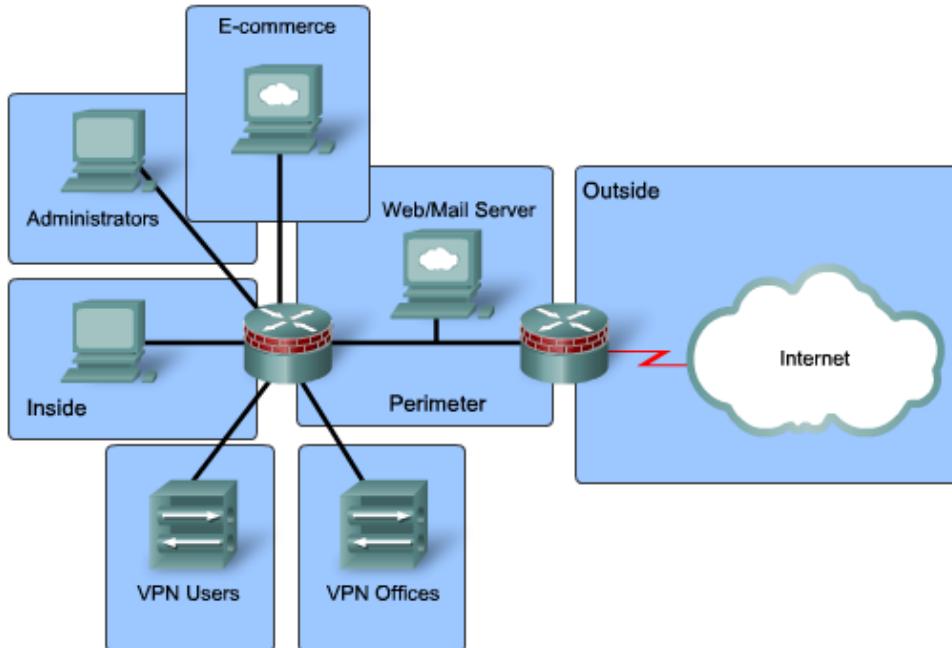


- *Firewalls redundantes*



ZPF: Arquitecturas típicas

- Arquitectura de *Firewall* compleja e perspectiva zonal

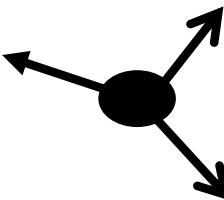


ZPF: Operação

- Acções possíveis (configuração assente no SDM)



Inspect



Drop

Equivalente ao “deny” das ACLs.
Pode ser complementado com a opção “log”.



Pass

Equivalente ao “permit”. Ao contrário do “inspect”, é uma acção stateless.

Equivalentes ao “ip inspect” do CBAC.
Automaticamente permite todo o tráfego associado sessão inspeccionada e a sessões complementares relacionadas (e.g., ftp, H.323).

ZPF: Configuração

- Acções possíveis
 - **inspect**
 - Abre a *firewall* ao tráfego de retorno e às mensagens ICMP que lhe poderão estar associadas. Nalguns protocolos como o FTP e o H.323 permite também a abertura de sessões adicionais em portos negociados.
 - **pass**
 - \cong **permit** das ACL. Abre passagem ao tráfego mas nada faz quanto aos pacotes retornados
 - **drop**
 - \cong **deny** das ACL. Descarta tráfego mencionado.

Ambas as opções permitem efectuar *rate limiting* para efeitos de DoS

ZPF: Operação

- Regras de utilização
 - As zonas devem ser configuradas antes de lhes associar interfaces
 - Se o tráfego tiver de fluir entre todas as interfaces do *router* cada interface deve ser membro de uma zona
 - Cada interface pode ser membro de uma única zona
 - Por omissão o tráfego flui entre interfaces da mesma zona
 - Apenas flui tráfego numa interface associada a uma zona se existir uma política “pass” ou “inspect” entre esta e outra zona
 - Não pode fluir tráfego entre uma interface integrada numa zona e uma interface não integrada em nenhuma zona
 - Interfaces não integradas em zonas podem usufruir do CBAC
 - Quando se pretende que uma interface não seja abrangida pelo ZPF pode ser necessário integrá-la numa zona e configurar uma política “pass-all” (\equiv *dummy policy*) entre esta e todas as zonas necessárias.



ZPF: Operação

- Políticas por omissão entre pares de zonas

The source policy application and default policy for traffic is applied according to these rules:

Source interface member of zone?	Destination interface member of zone?	Zone-pair exists?	Policy exists?	RESULT
NO	NO	N/A	N/A	No impact of zoning/policy
YES (zone 1)	YES (zone 1)	N/A*	N/A	No policy lookup (PASS)
YES	NO	N/A	N/A	DROP
NO	YES	N/A	N/A	DROP
YES (zone 1)	YES (zone 2)	NO	N/A	DROP
YES (zone 1)	YES (zone 1) (zone 2)	YES	NO	DROP
YES (zone 1)	YES (zone 2)	YES	YES	policy actions



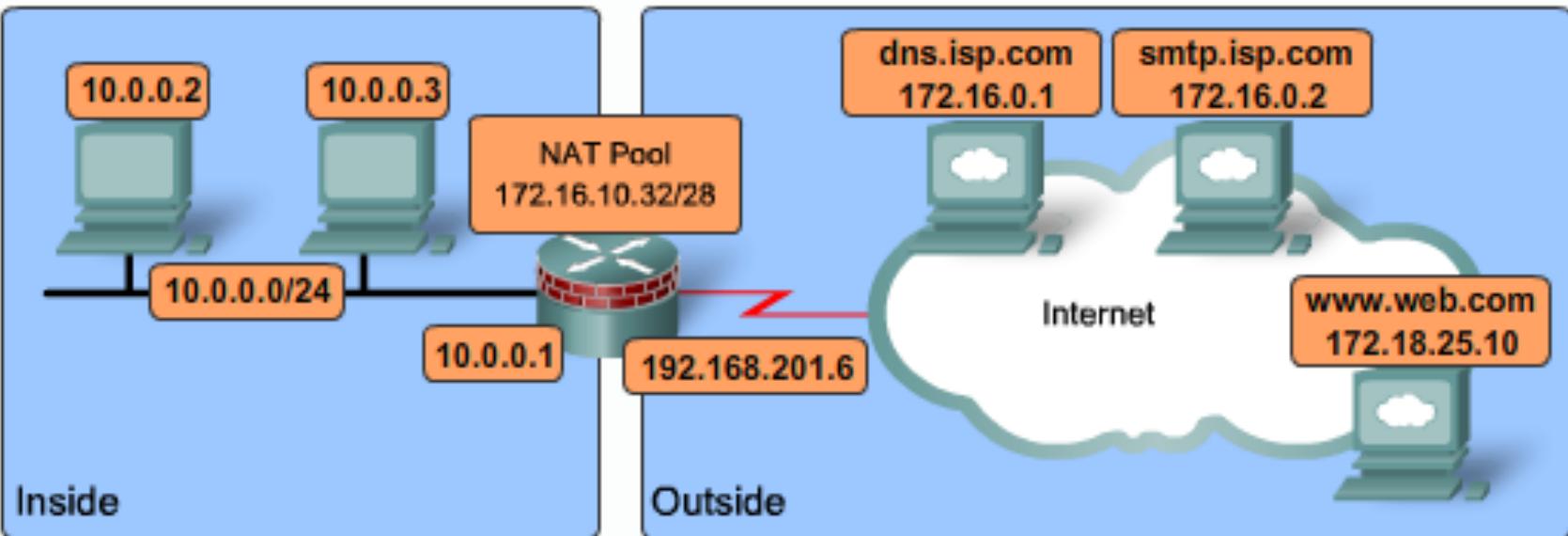
ZPF: Operação

- Se o *router* for um extremo da sessão as regras são diferentes
 - Todas as interfaces IP do *router* encontram-se integradas de forma tácita na “*self zone*”
 - Se não forem configuradas políticas para regular o tráfego entre uma qualquer zona e a *self zone*, este flui por omissão.
 - Políticas que envolvam a *self zone* apenas influenciam tráfego com origem ou destino no *router* e não o tráfego em trânsito

Source interface member of zone?	Destination interface member of zone?	Zone-pair exists?	Policy exists?	RESULT
ROUTER	YES	NO	N/A	PASS
ROUTER	YES	YES	NO	PASS
ROUTER	YES	YES	YES	policy actions
YES	ROUTER	NO	N/A	PASS
YES	ROUTER	YES	NO	PASS
YES	ROUTER	YES	YES	policy actions



ZPF: Configuração



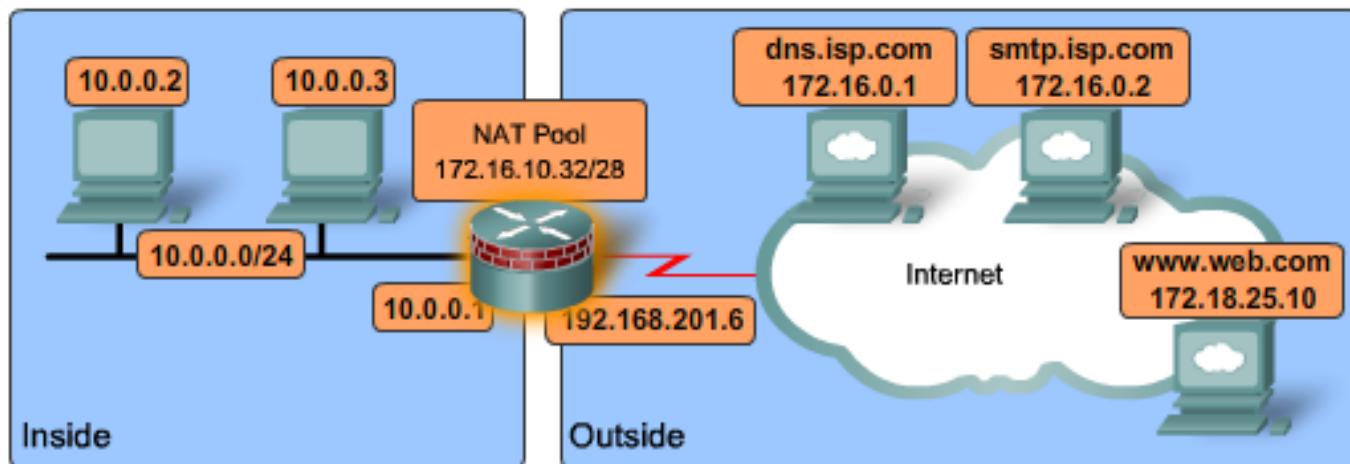
- 1 Create the zones.
- 2 Define traffic classes.
- 3 Define firewall policies.
- 4 Assign policy maps to zone pairs.
- 5 Assign router interfaces to zones.



ZPF: Configuração

1. Criação de zonas

```
Router(config)# zone security zone-name  
Router(config-sec-zone)# description line-of-description
```



```
FW(config)# zone security Inside  
FW(config-sec-zone)# description Inside network  
FW(config)# zone security Outside  
FW(config-sec-zone)# description Outside network
```



ZPF: Configuração

2. Definir classes de tráfego

- Nível 3 (IP) / Nível 4 (TCP/UDP/ICMP)

```
Router(config)# class-map type inspect [match-any | match-all] class-map-name
```

- Por omissão assume-se “match-any” (o tráfego tem que verificar apenas uma das condições definidas de seguida para integrar a classe)

- Nível 7 (para DPI de aplicações, usável apenas c/ *policy maps L7*)

```
Router(config)# class-map type inspect protocol-name [match-any | match-all] class-map-name
```

- Dentro da classe é identificado o tipo específico de tráfego
 - Com base em ACLs (*numbered* ou *named*)

```
Router(config-cmap)# match access-group {access-group | name access-group-name}
```

- Com base no tipo de protocolo

```
Router(config-cmap)# match protocol protocol-name
```

- Com base noutra classe de tráfego (i.e., definição encadeada)

```
Router(config-cmap)# match class-map class-map-name
```



ZPF: Configuração

2. Definir classes de tráfego (cont.)

- Para *class maps* L7 há cláusulas `match` específicas a cada protocolo.
- Por exemplo, para especificar tráfego HTTP que contém *applets java*:

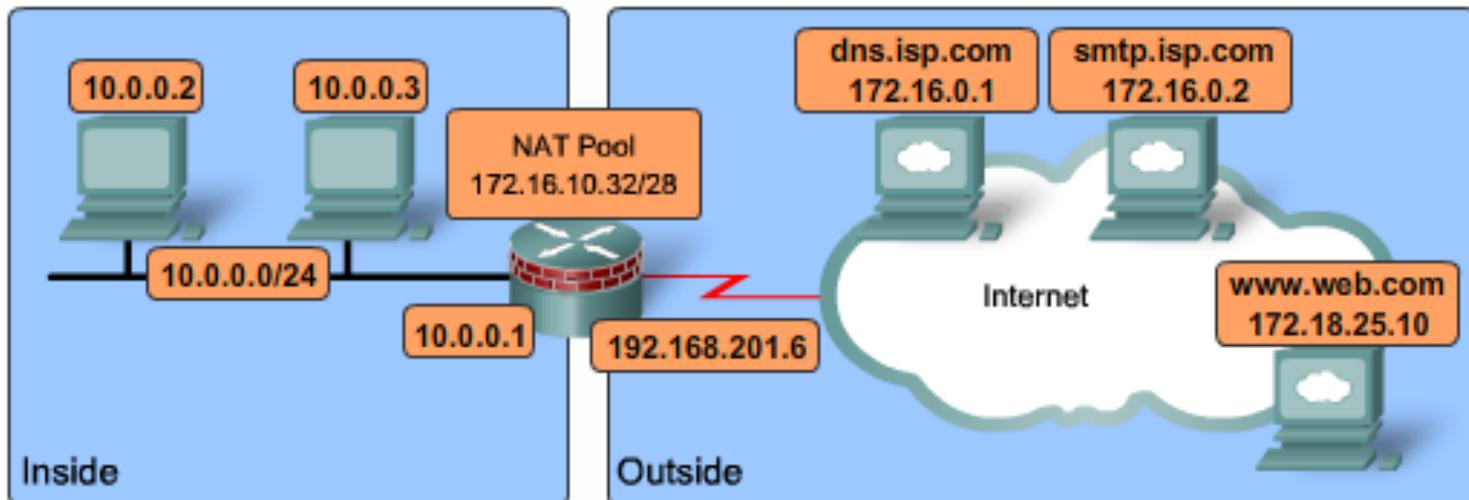
```
Router(config)# class-map type inspect http JAVA-APPLETS-CM  
Router(config-cmap)# match response body java
```

- Os *class maps* L7 devem ser associados a *policy maps* L7 para explorar os mecanismos de inspeção de tráfego da camada aplicacional.



ZPF: Configuração

2. (cont.) Definir classes de tráfego



```
FW(config)# class-map type inspect FOREXAMPLE  
FW(config-cmap)# match access-group 101  
FW(config-cmap)# exit  
FW(config)# access-list 101 permit ip 10.0.0.0 0.0.0.255 any
```



ZPF: Configuração

2. A sequência dos matches pode ser crítica

- Exemplo 1 (ordem correta)

```
Router(config)# class-map type inspect match-any MEU-CM  
Router(config-cmap)# match protocol http  
Router(config-cmap)# match protocol tcp
```

- Neste caso o tráfego HTTP apenas é tratado pelos mecanismos específicos da firewall para tratar fluxos HTTP porque é classificado pela primeira linha de match.
- Ou seja devemos organizar os classificadores do mais específico para o mais genérico.

- Exemplo 2 (ordem incorreta)

```
Router(config)# class-map type inspect match-any MEU-CM  
Router(config-cmap)# match protocol tcp  
Router(config-cmap)# match protocol http
```

- Neste caso o tráfego HTTP iria ser classificado como tráfego TCP e tratado pelos mecanismos de inspeção TCP!



ZPF: Configuração

3. Especificar as políticas da *firewall*

- Criação de uma política

```
Router(config)# policy-map type inspect [{{http|im|imap|p2p|pop3|smtp|sunrpc}}]  
policy-map-name
```

- Classe(s) de tráfego a considerar nesta política

```
Router(config-pmap)# class type inspect class-name
```

- O restante tráfego pode ser especificado da seguinte forma

```
Router(config-pmap)# class class-default
```

- Finalmente a acção a aplicar ao tráfego identificado

```
Router(config-pmap-c)# pass | inspect | drop | {police rate bps burst bytes} |  
{service-policy {http|im|imap|p2p|pop3|smtp|sunrpc} {DPI-policy-map-name}}  
{urlfilter URLFilter-map-name }  
[log]
```

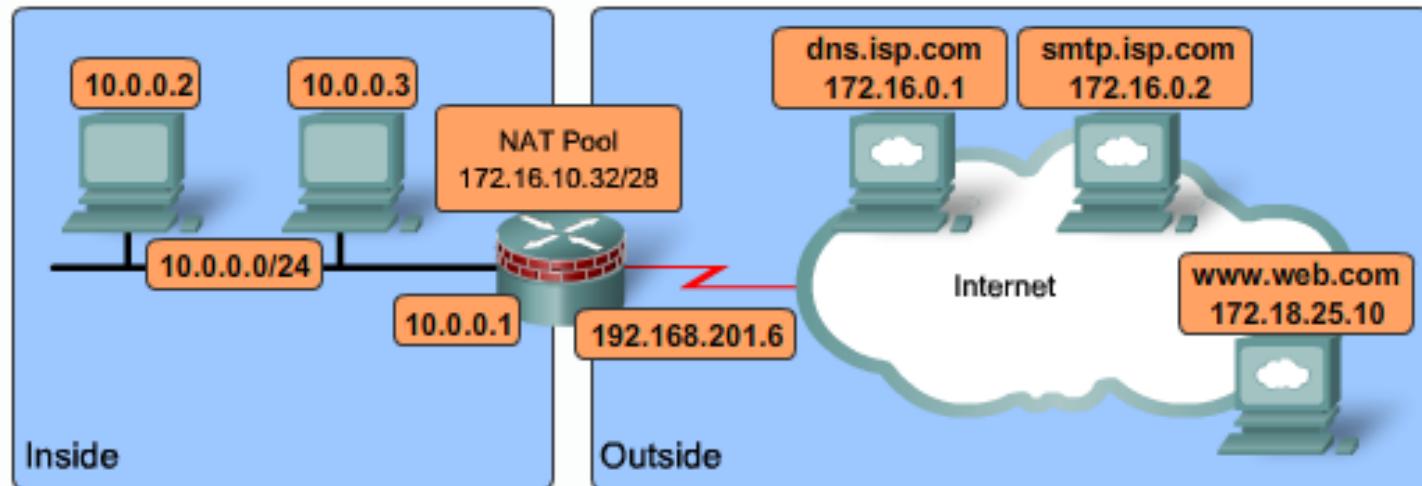
im: AOL, ICQ, MSN Messenger,
Messenger, Yahoo Messenger

- O **service-policy** permite juntar uma política de *Deep Packet Inspection* (DPI) a qualquer política *top-level*



ZPF: Configuração

3. (cont.) Especificar as políticas da *firewall*



```
FW(config)# policy-map type inspect InsideToOutside  
FW(config-pmap)# class type inspect FOREXAMPLE  
FW(config-pmap-c)# inspect
```

O **inspect** abre a *firewall* ao tráfego de retorno e às mensagens ICMP que lhe poderão estar associadas.



ZPF: Configuração

4. Aplicar a política da *firewall*

- Criar o par de zonas pretendido

```
Router(config)# zone-pair security zone-pair-name source [source-zone-name |  
self] destination [self | destination-zone-name]
```

- Aplicar-lhe a política definida

```
Router(config-sec-zone-pair)# service-policy type inspect policy-map-name
```

5. Integrar uma interface numa zona de segurança

```
Router(config)# interface ifname  
Router(config-if)# zone-member security zone-name
```



ZPF: Troubleshooting

- *Composição de zonas*

```
Router# show zone security [<zone-name>]
```

- Pares de zonas formados

```
Router# show zone-pair security [source <source-zone-name>] [destination  
<destination-zone-name>]
```

- Mapas de políticas

```
Router# show policy-map type inspect [<policy-map-name>] [class <class-map-  
name>]
```

- Estatísticas de aplicação das políticas

```
Router# show policy-map type inspect zone-pair [<zone-pair-name>] [sessions]
```

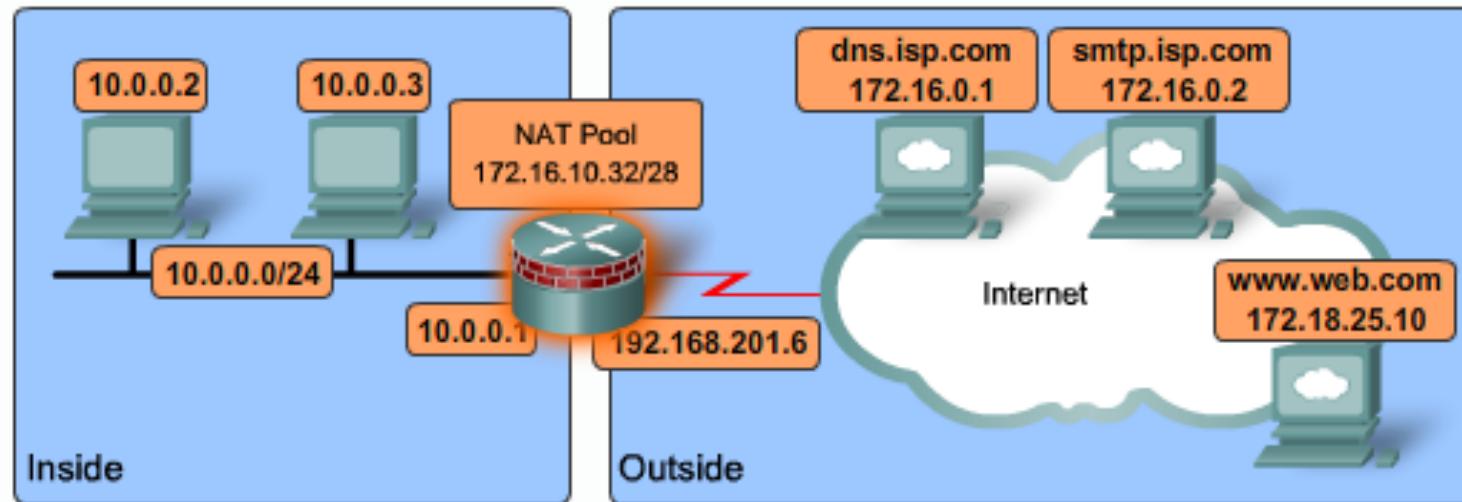
- Estatísticas mais detalhadas

```
Router# show policy-map type inspect inspect { <policy name> [class <class  
name>] | zone-pair [<zone-pair name>] [sessions | urlfilter cache] }
```



ZPF: Configuração

4. (cont) Aplicar a política da *firewall*



```
FW(config)# zone-pair security InsideToOutside source Inside destination Outside
FW(config-sec-zone-pair)# description Internet Access
FW(config-sec-zone-pair)# service-policy type inspect InsideToOutside
FW(config-sec-zone-pair)# interface F0/0
FW(config-if)# zone-member security Inside
FW(config-if)# interface S0/0/0.100 point-to-point
FW(config-if)# zone-member security Outside
```



ZPF: Configuração de aspectos avançados

- *Rate Policing*
 - IOS \geq 12.4(9)T
 - Ao contrário dos mecanismos de *rate policing* por interface, neste caso apenas se emprega a política de **pass** para o tráfego que se encontra dentro do limite e **drop** para o restante.

```
policy-map type inspect private-allowed-policy
  class type inspect http-class
    inspect
    police rate bps rate bytes
```

```
[8000-2000000000]
[1000-5120000000]
```

- Nota: Estas políticas podem ser aplicadas em simultâneo com políticas de *rate limiting* por interface mas devem ser coerentes com aquelas.



ZPF: Configuração de aspectos avançados

- *Session Control*

- Controlo do número de sessões
- Trata-se de um mecanismo de protecção DoS extra que pode ser configurado por classe de tráfego.
- Apenas disponível para accções do tipo **inspect**

```
R(config)#parameter-map type inspect my-parameter-map
R(config-profile)#sessions maximum num
                                     ↳ [1-2147483647]
policy-map type inspect private-allowed-policy
  class type inspect http-class
    inspect my-parameter-map
```



ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC)*
 - Protocolos L7: http, im, imap, p2p, pop3, smtp, sunrpc
 - **im**: AOL, ICQ, MSN Messenger, Messenger, Yahoo Messenger
 - **p2p**: BitTorrent, eDonkey, FastTrack, Gnutella, KaZaA / KaZaA2, WinMX
 - Permite restringir tipos de mensagens e conteúdos potencialmente vulneráveis nos protocolos mencionados
 - Esta abordagem denomina-se *Deep Packet Inspection (DPI)*

O IOS efectua o reconhecimento das aplicações P2P através da tecnologia **Network-Based Application Recognition (NBAR)**. O NBAR representa heurísticas de rede (PDLMs: *Protocol Description Language Module*) actualizáveis (`ip nbar pdlm <file-location>`) de modo a que a detecção acompanhe a evolução das aplicações P2P.



ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC)*
 - Os *class maps* L7 apenas podem ser usadas com *policy maps* L7
 - Os *policy maps* L7 (i.e., *application-specific*) não podem ser aplicados diretamente a uma zona. Têm de ser usados como “filhos” de *policy maps* L3/L4.
 - Estratégia de configuração
 1. Configuração de *class-maps* e *policy-maps* específicos a cada aplicação
 2. Aplicação desses *maps* aos *inspection maps* existentes



ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC)*
 - Políticas de inspeção disponíveis
 - HTTP
 - Limitação do tamanho das transferências
 - Limitação do tamanho dos URL
 - Limitação do tipo de conteúdos
 - SMTP
 - Limitação do tamanho dos conteúdos
 - Actividade protocolar em conformidade com as normas
 - POP3, IMAP
 - Forçar utilizadores a usar mecanismos de autenticação corretos



ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC) - Exemplos*

```
class-map type inspect http match-any http-aic-cmap
  match request port-misuse any
  match req-resp protocol-violation } Actividades não permitidas
  match request body length gt 4096
```

```
policy-map type inspect http http-aic-pmap
  class type inspect http http-aic-cmap
    reset } Reacção sobre as sessões indesejadas
    log
```

```
class-map type inspect match-any http-cmap
  match protocol http
```

```
class-map type inspect match-any other-traffic-cmap
  match protocol smtp
  match protocol dns
  match protocol ftp
```

```
policy-map type inspect priv-pub-pmap
  class type inspect http-cmap
    inspect
    service-policy http http-aic-pmap
  class type inspect other-traffic-cmap
    inspect
```

Sobre sessões HTTP fazer vigilância stateful (1) e controle (reset+log) sobre as sessões indesejadas (2).

Sobre as outras sessões de interesse fazer apenas vigilância stateful.

ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC) - HTTP*
 - Conteúdo de cabeçalho (*Header inspection*)

```
APPFW-6-HTTP_HDR_REGEX_MATCHED
```

```
match {request|response|req-resp} header regex parameter-map-name
```

```
parameter-map type regex non_ascii_regex  
pattern "[^\x00-\x80]"
```

```
class-map type inspect http non_ascii_cm  
match req-resp header regex non_ascii_regex
```

```
policy-map type inspect http non_ascii_pm  
class type inspect http non_ascii_cm  
reset
```



ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC) - HTTP*

- *Header inspection*
- *Header count inspection*
- *Header field inspection*
- *Header field length inspection*
- *Header field repetition*
- *Method inspection*
- *URI inspection*
- *URI length inspection*
- Argument inspection
- Argument length inspection
- Body inspection
- Body (Content) length inspection
- Status line inspection
- Content-type inspection
- Port-misuse inspection
- Strict-HTTP inspection
- Transfer-Encoding inspection
- Java Applet inspection

[Zone-Based Policy Firewall
Design and Application
Guide](#)

ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC)* - HTTP
 - Server-based URL Filtering

```
parameter-map type urlfilter websense-parmap
  server vendor [n2h2 | websense] 10.1.1.1
```

- URL white and black lists

```
parameter-map type urlfilter websense-parmap
  exclusive-domain deny .disallowed.com
  exclusive-domain permit .cisco.com
```

```
class-map type inspect match-any http-cmap
  match protocol http
```

```
policy-map type inspect http-filter-pmap
  class type inspect http-cmap
    inspect
    urlfilter websense-parmap
```

Sobre sessões HTTP fazer
vigilância stateful e
filtragem de URLs.

ZPF: Configuração de aspectos avançados

- *Application Inspection and Control (AIC) - P2P*
 - Configuração

```
class-map type inspect match-any my-p2p-class
  match protocol [bittorrent | edonkey | fasttrack | gnutella |
    kazaa | kazaa2 | winmx ]
    [signature (optional)]
```

```
policy-map type inspect private-allowed-policy
  class type inspect my-p2p-class
    [drop | inspect | pass]
```

Application	Native Ports (as recognized by 12.4(15)T PAM list)
bittorrent	TCP 6881-6889
edonkey	TCP 4662
fasttrack	TCP 1214
gnutella	TCP 6346-6349 TCP 6355,5634 UDP 6346-6348
kazaa2	Dependent on PAM
winmx	TCP 6699

- A utilização de assinaturas configuráveis fornece ao NBAR pistas adicionais de classificação de tráfego P2P que ajudam a acompanhar o *port-hopping* típico destas aplicações.
- Esta estratégia contudo tem custos acrescidos ao nível da CPU e ritmo de processamento do tráfego.

ZPF: Configuração de aspectos avançados

- Zona *self* - especificidades

- A inspecção de tráfego ao nível aplicacional não está disponível
- A limitação do número e débito de sessões não está disponível
- Exemplo (gestor SNMP: 172.17.100.11, servidor TFTP: 172.17.100.17)

```
class-map type inspect match-any self-service-cmap
  match protocol tcp
  match protocol udp
  match protocol icmp
  match protocol h323

class-map type inspect match-all to-self-cmap
  match class-map self-service-cmap
  match access-group 120

class-map type inspect match-all from-self-cmap
  match class-map self-service-cmap

class-map type inspect match-all tftp-in-cmap
  match access-group 121

class-map type inspect match-all tftp-out-cmap
  match access-group 122
```



ZPF: Configuração de aspectos avançados

- Zona *self* - especificidades
 - Exemplo (cont.):

```
policy-map type inspect to-self-pmap
  class type inspect to-self-cmap
    inspect
  class type inspect tftp-in-cmap
    pass

policy-map type inspect from-self-pmap
  class type inspect from-self-cmap
    inspect
  class type inspect tftp-out-cmap
    pass

zone security private
zone security internet
zone-pair security priv-self source private destination self
  service-policy type inspect to-self-pmap
zone-pair security net-self source internet destination self
  service-policy type inspect to-self-pmap
zone-pair security self-priv source self destination private
  service-policy type inspect from-self-pmap
zone-pair security self-net source self destination internet
  service-policy type inspect from-self-pmap
```



ZPF: Configuração de aspectos avançados

- Zona *self* - especificidades
 - Exemplo (conclusão):

```
interface FastEthernet 0/0
  ip address 172.16.100.10
  zone-member security internet

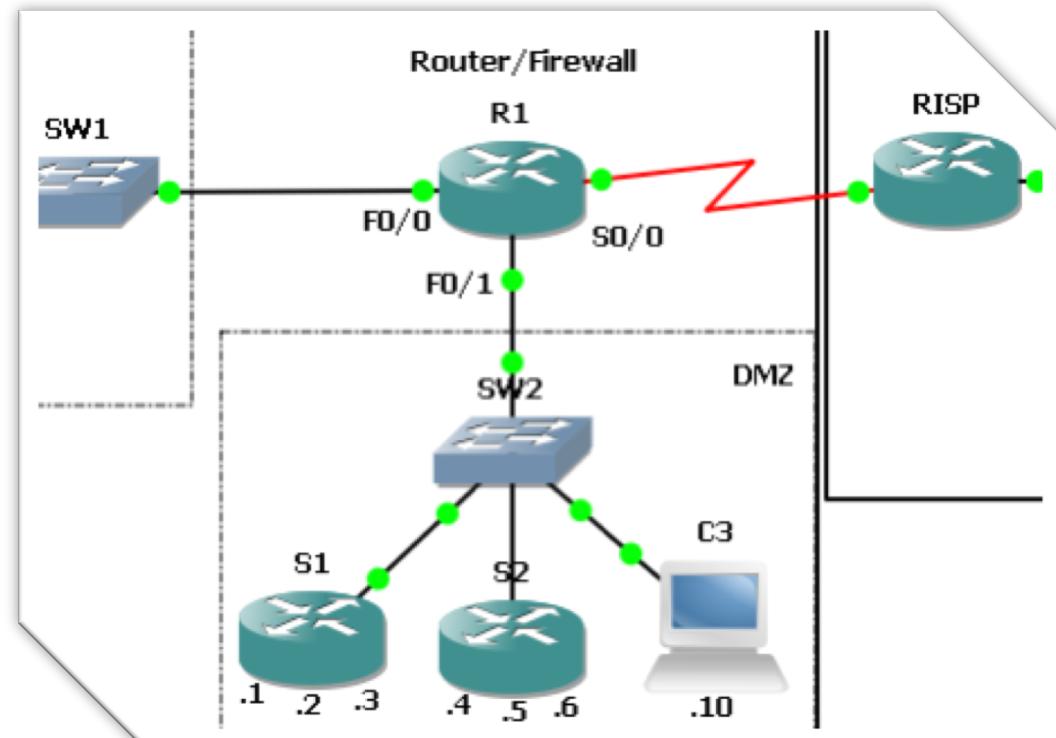
interface FastEthernet 0/1
  ip address 172.17.100.10
  zone-member security private

access-list 120 permit icmp 172.17.100.0 0.0.0.255 any
access-list 120 permit icmp any host 172.17.100.10 echo
access-list 120 deny icmp any any
access-list 120 permit tcp 172.17.100.0 0.0.0.255 host 172.17.100.10 eq www
access-list 120 permit tcp any any eq 443
access-list 120 permit tcp any any eq 22
access-list 120 permit udp any host 172.17.100.10 eq snmp
access-list 121 permit udp host 172.17.100.17 host 172.17.100.10
access-list 122 permit udp host 172.17.100.10 host 172.17.100.17
```



ZPF: Exercício

- Programe R1 de modo que a) a intranet possa estabelecer qualquer sessão com a DMZ/Internet; b) a DMZ aceite igualmente sessões da Internet mas apenas nos serviços públicos que mantém; c) da DMZ possam ser estabelecidas quaisquer sessões apenas para a Internet.



ZPF: Solução

1. Assegurar que R1-Firewall não possui qualquer configuração CBAC (Stop → Start)
2. Testar a conectividade entre a intranet e a DMZ/Internet

```
VPCS[1]> ping 194.65.52.10
194.65.52.10 icmp_seq=1 timeout
194.65.52.10 icmp_seq=2 ttl=63 time=32.000 ms
194.65.52.10 icmp_seq=3 ttl=63 time=28.000 ms
194.65.52.10 icmp_seq=4 ttl=63 time=62.000 ms
194.65.52.10 icmp_seq=5 ttl=63 time=61.000 ms
```

```
VPCS[1]> 2
VPCS[2]> ping 4.4.4.4
4.4.4.4 icmp_seq=1 ttl=62 time=36.000 ms
4.4.4.4 icmp_seq=2 ttl=62 time=26.000 ms
4.4.4.4 icmp_seq=3 ttl=62 time=28.000 ms
4.4.4.4 icmp_seq=4 ttl=62 time=30.000 ms
4.4.4.4 icmp_seq=5 ttl=62 time=33.000 ms
```



ZPF: Solução

3. Criar zonas de segurança

```
R1-Firewall#configure terminal
R1-Firewall(config)#zone security INTRANET
R1-Firewall(config-sec-zone)# description intranet da empresa
R1-Firewall(config-sec-zone)#
R1-Firewall(config-sec-zone)#zone security INTERNET
R1-Firewall(...)#${description} Public Internet (directly connected to ISP)
R1-Firewall(config-sec-zone)#
R1-Firewall(config-sec-zone)#zone security DMZ
R1-Firewall(config-sec-zone)#description DMZ (Internet servers and proxies)
R1-Firewall(config-sec-zone)#
R1-Firewall(config-sec-zone)#interface F0/0
R1-Firewall(config-if)#zone-member security INTRANET
R1-Firewall(config-if)#interface S0/0
R1-Firewall(config-if)#
R1-Firewall(config-if)#zone-member security INTERNET
R1-Firewall(config-if)#interface F0/1
R1-Firewall(config-if)#
R1-Firewall(config-if)#zone-member security DMZ
R1-Firewall(config-if)^z
```



ZPF: Solução

4. Verificar que qualquer sessão passa agora a ser proibida

```
VPCS[2]> 1
VPCS[1]> ping 194.65.52.10
194.65.52.10 icmp_seq=1 timeout
194.65.52.10 icmp_seq=2 timeout
194.65.52.10 icmp_seq=3 timeout
194.65.52.10 icmp_seq=4 timeout
194.65.52.10 icmp_seq=5 timeout

VPCS[1]> 2
VPCS[2]> ping 4.4.4.4
4.4.4.4 icmp_seq=1 timeout
4.4.4.4 icmp_seq=2 timeout
4.4.4.4 icmp_seq=3 timeout
4.4.4.4 icmp_seq=4 timeout
4.4.4.4 icmp_seq=5 timeout
```

4. Excepto sessões entre terminais da mesma interface
 4. Exemplo: PC1-PC2



ZPF: Solução

5. Configurar um police-map que envolva todas as sessões IP

```
R1-Firewall(config)#policy-map type inspect INTRANET_INITIATED-POLICY_MAP
R1-Firewall(config-pmap)# class class-default
R1-Firewall(config-pmap-c)# inspect
%No specific protocol configured in class class-default for inspection. All
protocols will be inspected
R1-Firewall(config-pmap-c)#exit
R1-Firewall(config-pmap)#exit
R1-Firewall(config)#zone-pair security INTRANET-DMZ-ZONE_PAIR source INTRANET
destination DMZ
R1-Firewall(...)#service-policy type inspect INTRANET_INITIATED-POLICY_MAP
R1-Firewall(config-sec-zone-pair)#^z
```

6. Consultar as sessões vigiadas pela ZPF

```
R1-Firewall#show policy-map type inspect zone-pair sessions
Zone-pair: INTRANET-DMZ-ZONE_PAIR
    Inspect
    Service-policy inspect : INTRANET_INITIATED-POLICY_MAP
        Class-map: class-default (match-any)
            Match: any
```



ZPF: Solução

7. Confirmar que da Intranet já é possível comunicar com a DMZ

```
VPCS[1]> ping 194.65.52.10
194.65.52.10 icmp_seq=1 ttl=63 time=52.000 ms
194.65.52.10 icmp_seq=2 ttl=63 time=95.000 ms
194.65.52.10 icmp_seq=3 ttl=63 time=29.000 ms
194.65.52.10 icmp_seq=4 ttl=63 time=43.000 ms
194.65.52.10 icmp_seq=5 ttl=63 time=17.000 ms
```

8. Consultar as sessões vigiadas pela ZPF

```
R1-Firewall#show policy-map type inspect zone-pair sessions
Zone-pair: INTRANET-DMZ-ZONE_PAIR
    Inspect
        Established Sessions
            Session 64850E50 (192.168.1.1:8)=>(194.65.52.10:0) icmp SIS_OPEN
                Created 00:00:03, Last heard 00:00:00
                    ECHO request
                    Bytes sent (initiator:responder) [80:80]
Service-policy inspect : INTRANET_INITIATED-POLICY_MAP

    Class-map: class-default (match-any)
        Match: any
```



ZPF: Solução

9. Confirmar que da intranet ainda não é possível ir à Internet

```
VPCS[2]> ping 4.4.4.4
4.4.4.4 icmp_seq=1 timeout
4.4.4.4 icmp_seq=2 timeout
4.4.4.4 icmp_seq=3 timeout
4.4.4.4 icmp_seq=4 timeout
4.4.4.4 icmp_seq=5 timeout
```



ZPF: Desafio

- Terminar o exercício
 - TPC



Referências

- 📖 CCNA Security Ch. 4 - Implementing Firewall Technologies
- 📖 Cisco IOS Security Command Reference - Release 12.4, 2008, Cisco Systems
- 📖 Cisco IOS Security Configuration Guide - Release 12.4, 2008, Cisco Systems



Obrigado pela atenção. Alguma dúvida?

