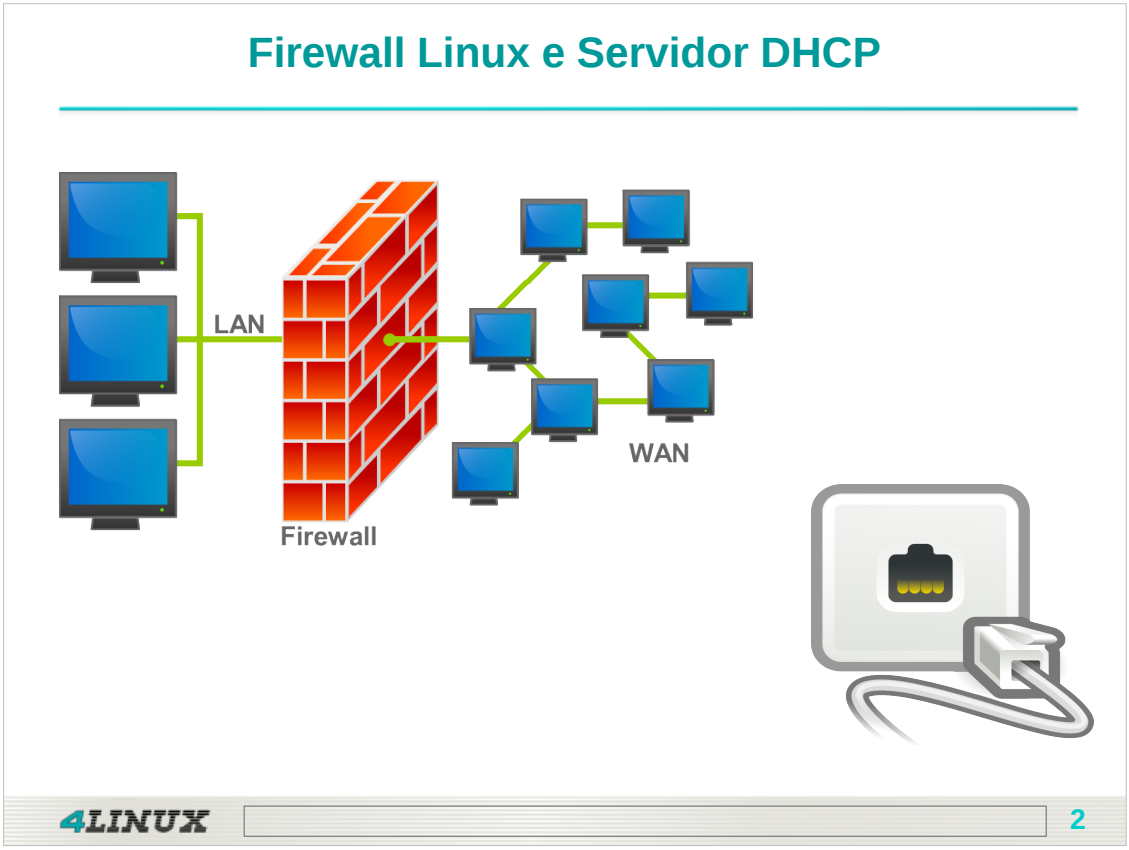
Curso 452

Linux Security Servers in Cloud

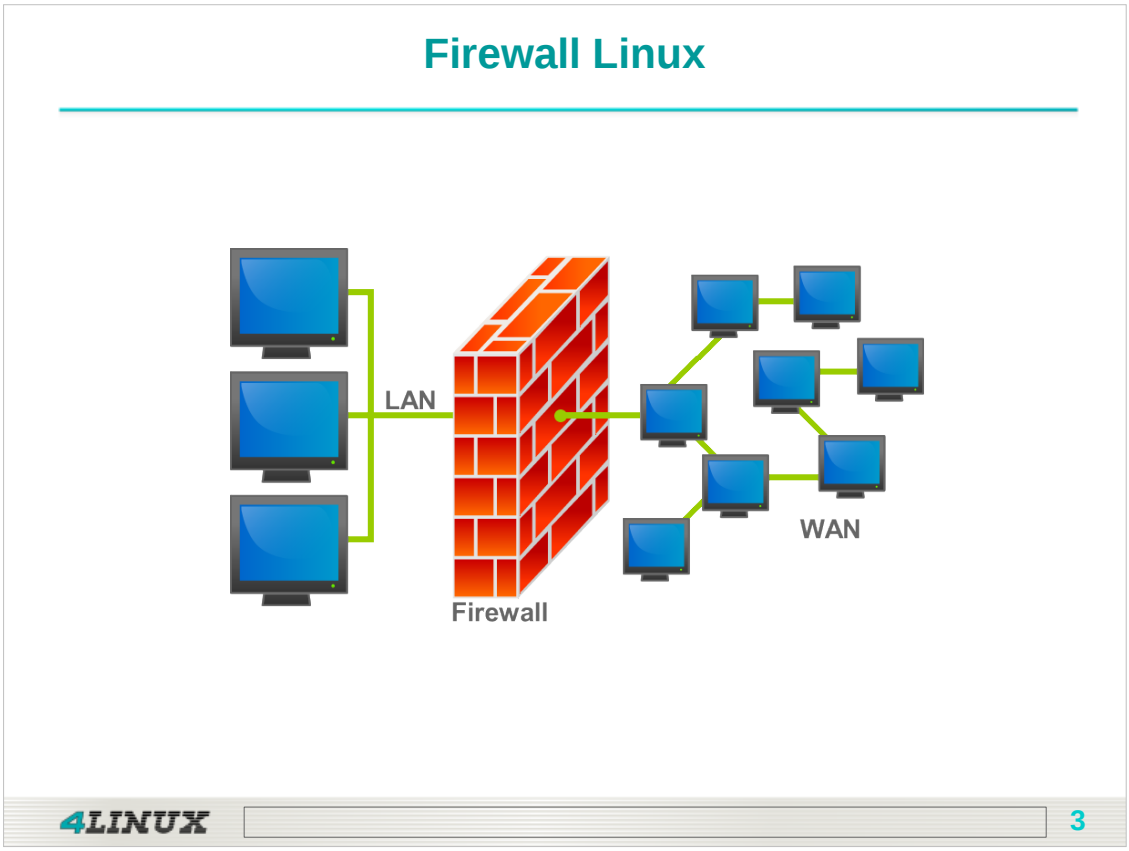


Cenário

A fim de que ninguém acesse dados indevidos da empresa ou mesmo de seus clientes, a empresa **DEXTER COURIER** deve contar com um excelente Firewall para bloquear tentativas de invasão e liberar apenas o tráfego necessário aos serviços que ficam na máquina **“DMZ”.**

Proposta de solução

Faremos a implementação do Firewall baseado no Iptables, gerenciando as políticas de entrada de pacotes na empresa Dexter, bem como a liberação de acesso externo ao servidor da DMZ.



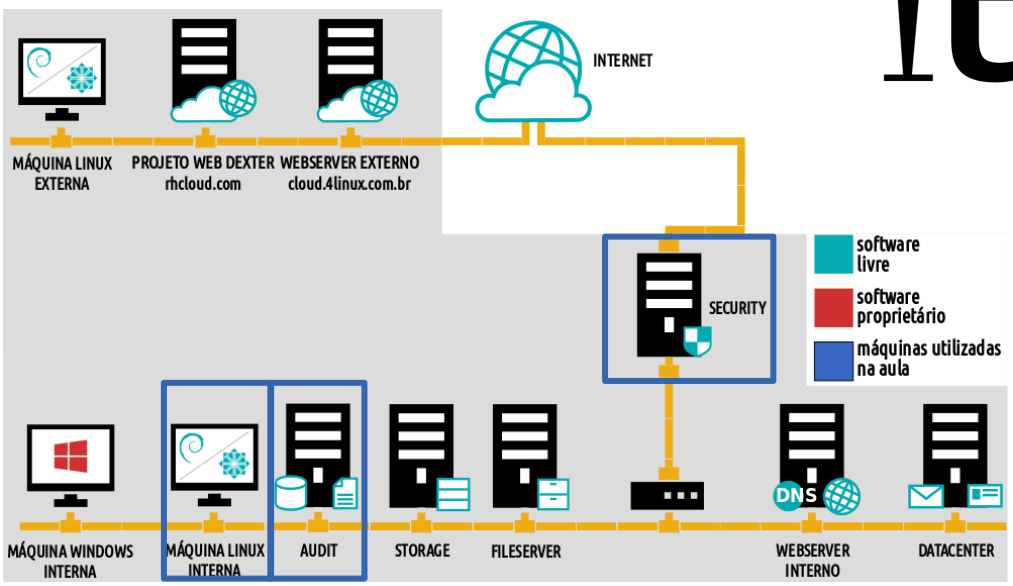
Cenário

A fim de que ninguém acesse dados indevidos da empresa ou mesmo de seus clientes, a empresa **DEXTER COURIER** deve contar com um excelente Firewall para bloquear tentativas de invasão e liberar apenas o tráfego necessário aos serviços que ficam na máquina **“DMZ”.**

Proposta de solução

Faremos a implementação do Firewall baseado no Iptables, gerenciando as políticas de entrada de pacotes na empresa Dexter, bem como a liberação de acesso externo ao servidor da DMZ.

IT Experience



Objetivos da Aula

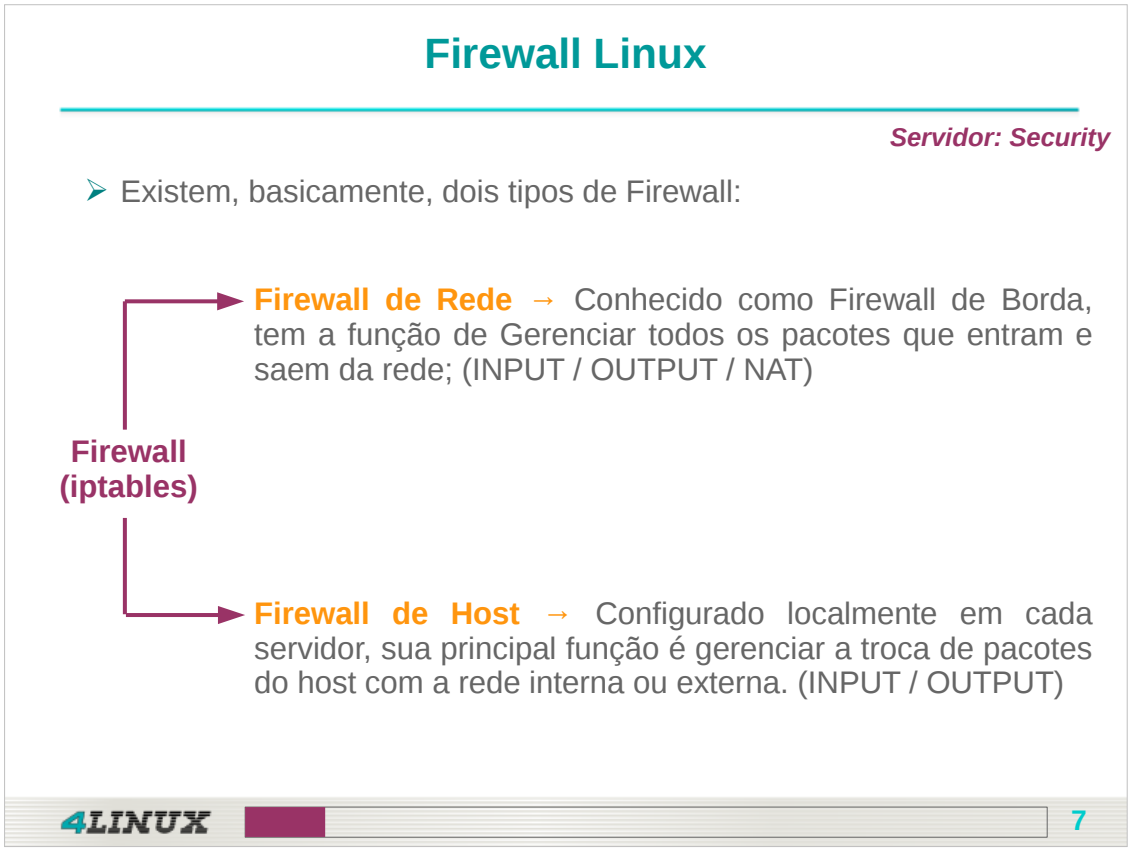
Aula 02 – Servidor Firewall (parte 1/2)

- Introdução Teórica;
- Conhecer a infraestrutura da empresa;
- Conhecer os tipos de tabelas do Iptables;
- Compreendendo as políticas e as exceções;
- Implementação prática com Script de Firewall.

Objetivos da Aula

Aula 02 – Servidor DHCP (parte 2/2)

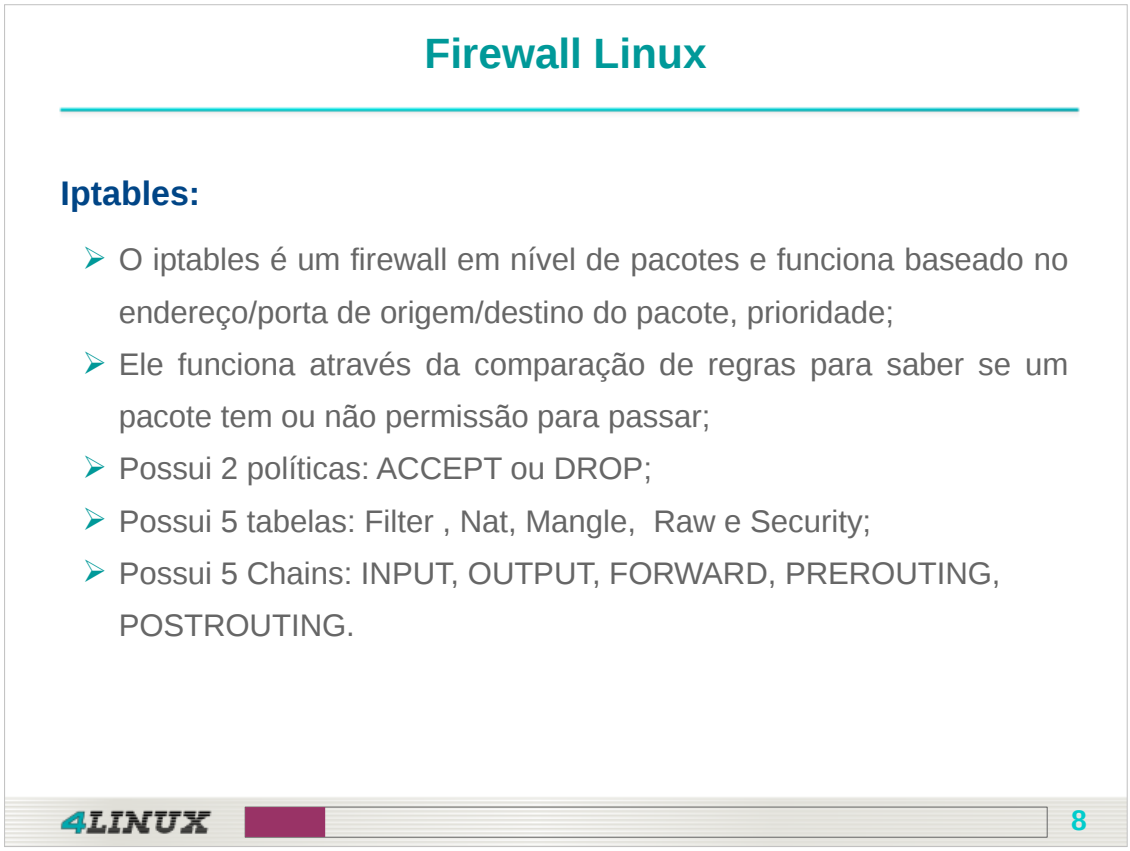
- Introdução Teórica;
- Entender o funcionamento do serviço;
- Implementar na prática o Servidor DHCP;
- Configurar Servidor DHCP;
- Configurar clientes DHCP;
- Fixar IP via DHCP.



O que é um Firewall?

Um firewall faz o filtro de pacotes que passam na rede. Para configurar um firewall é necessário o conhecimento sobre a estrutura da rede em questão e dos diferentes protocolos envolvidos na comunicação, isto é, dos serviços que a rede usa para que eles não percam a comunicação.

O objetivo em ter uma máquina fazendo o papel de Firewall Gateway em nossa é rede é minimizar as tentativas de ataques que elas recebem, tentando impedir possíveis invasões e levantamento de informações.



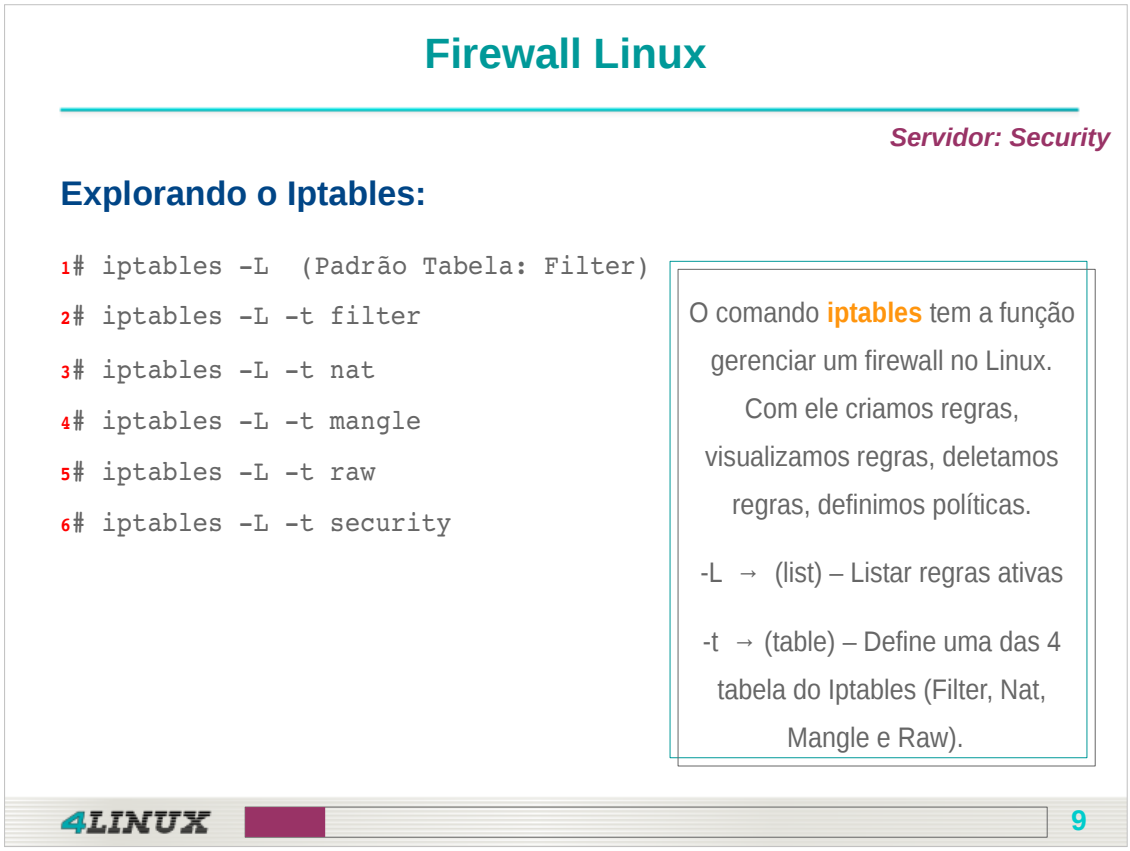
Iptables

Os sistema GNU/Linux com Kernel 2.6 trabalham com o "Iptables" para fazer o gerenciamento de regras de Firewall. Lembrando que o "Iptables" é apenas um "frontend" que gerencia o suporte "Netfilter" no "Kernel".

O que são as CHAINS?

Basicamente no iptables as regras são organizadas em tabelas, essas tabelas possuem o que chamamos de CHAINS. É nas CHAINS que são definidas as regras para o nosso "firewall" sendo que uma CHAIN deverá possuir também uma política padrão (drop ou accept).

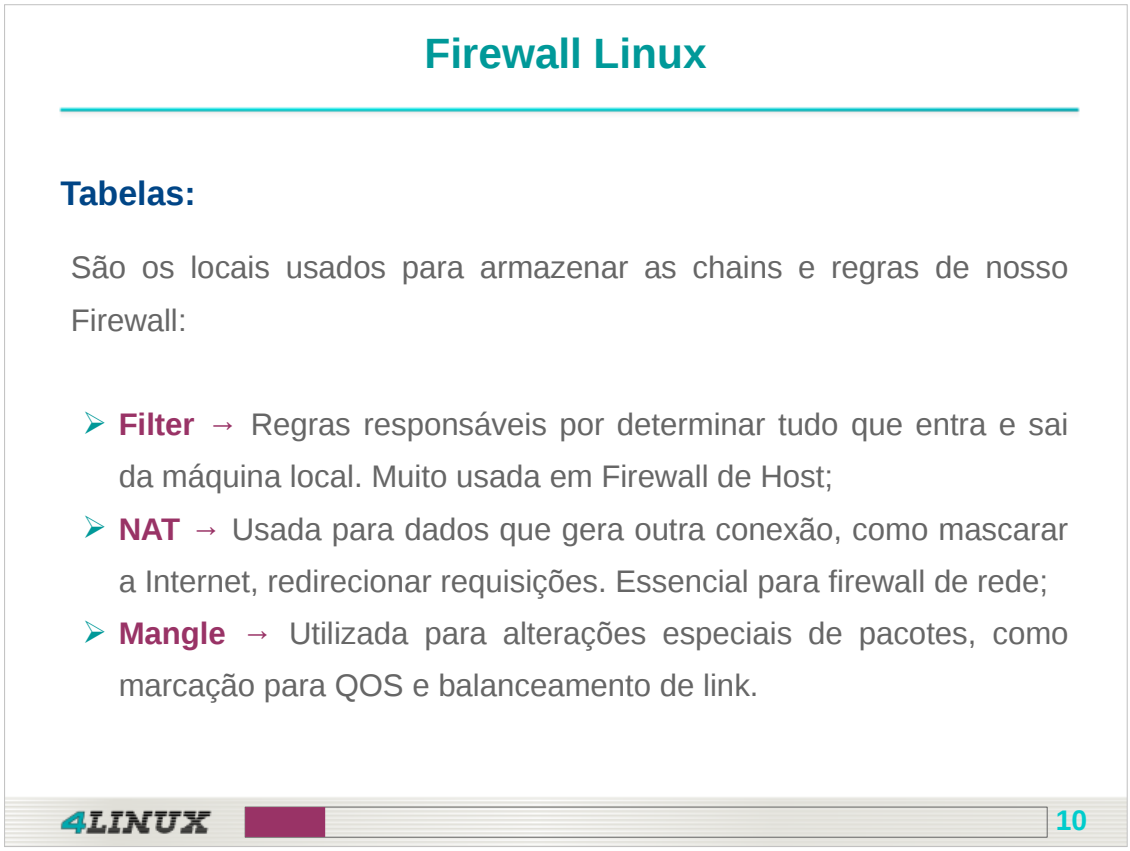
A tradução literal para CHAIN seria "correntes", assim cada tabela teria uma corrente, onde cada elo corresponderia a uma regra.



Características do iptables :

Filtro de pacotes statefull → Isso significa que o iptables é capaz de atuar sobre as camadas do protocolo TCP;

Modularidade → A configuração do kernel é modular e com o netfilter não é diferente, pois novas funcionalidades podem ser adicionadas sem muito esforço. Um módulo só será usado se for da necessidade do administrador;



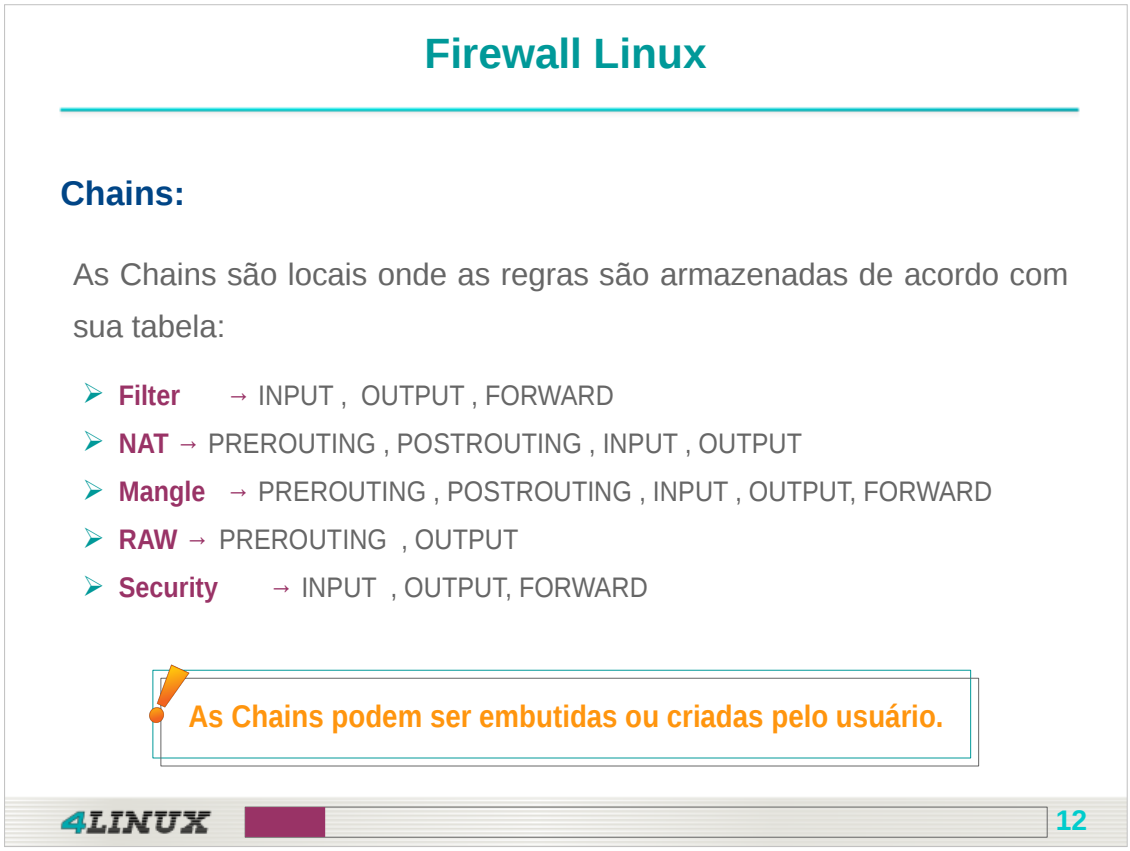
Revisando os conceitos de Rede:

Os dados são transmitidos na Internet agrupados em pacotes TCP (de maneira geral). Esses pacotes TCP podem conter até 1460 bytes de dados. Além dos dados, 40 bytes adicionais vão junto no pacote. Nesses 40 bytes a mais seguem:

→ **IP de origem**
→ **IP de destino**
→ **Porta de origem**
→ **Porta de destino**
→ **Códigos de verificação**
→ **Número do pacote**

Revisando os conceitos de Rede:

Os códigos de verificação servem para garantir a integridade dos dados que estão sendo trafegados na rede. A função básica do IP é cuidar do endereçamento e entrega de pacotes. A função básica do TCP é fazer verificações de erros e numeração de portas. Logo, os dados serão transmitidos de forma quebrada, em pacotes menores.



Revisando os conceitos de Rede:

Existem 65.536 portas TCP e UDP. Elas são numeradas de 0 e 65.535. As portas baixas estão na faixa entre 0 a 1023. Elas estão reservadas para serviços mais conhecidos como: servidor web, ftp, ssh, telnet, servidores de e-mail, compartilhamento de arquivos, como, por exemplo, Samba, NFS etc. Portas altas estão faixa acima de 1023.

Firewall Linux

Política e Regras:

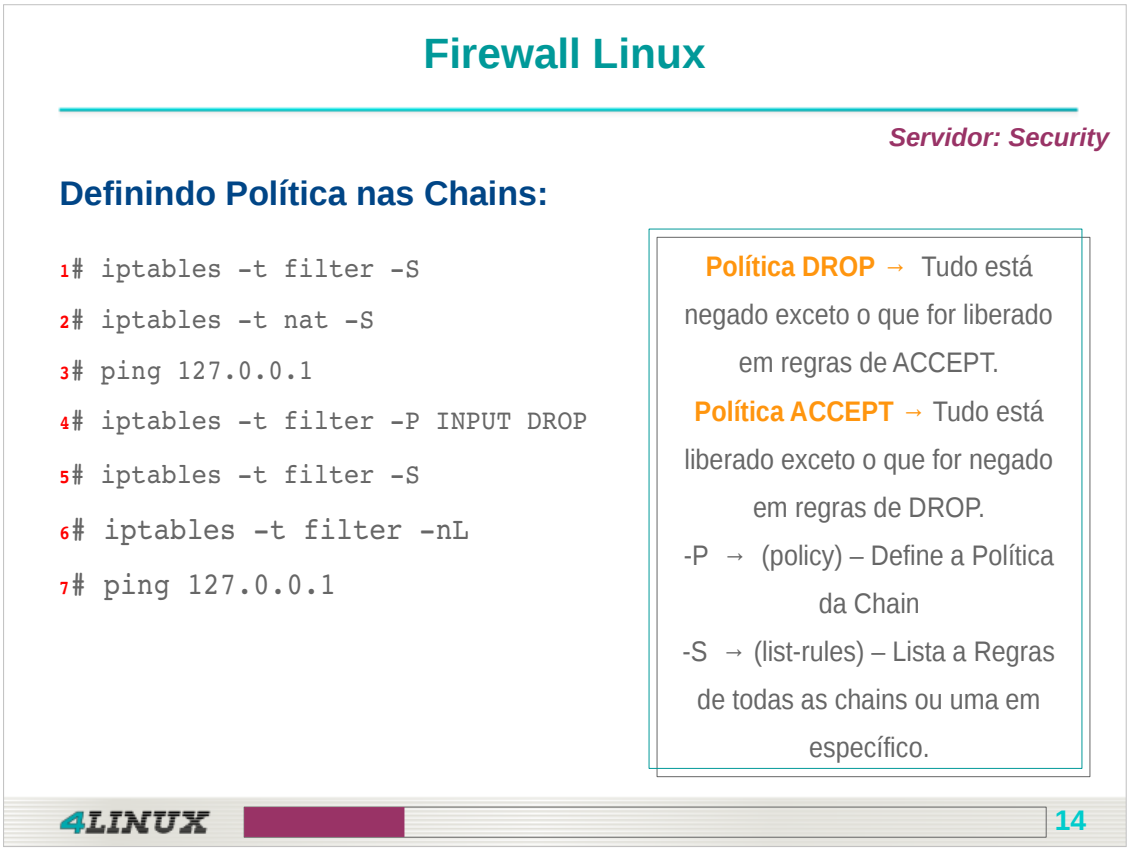
Cada Chain possui uma política padrão que vai determinar que tipo de regras você irá criar na Chain:

- **Política DROP** → Cria-se regras de Bloqueio de Pacotes;
- **Política ACCEPT** → Cria-se regras de Liberação de Pacotes;

Qual política é mais eficaz?
É mais fácil saber tudo o que precisa ser bloqueado ou tudo que minha empresa precisa que seja liberado?

Compreendendo as políticas e as exceções:

A metodologia utilizada para implementação do “firewall“ será a seguinte:



Compreendendo as políticas e as exceções:

Iremos negar todo o tráfego para as "chains" de "INPUT", "OUTPUT" e "FORWARD" da tabela "filter", posteriormente iremos definir a relação dos serviços que devem ser liberados no "firewall", as chamadas exceções. Todo o tráfego de pacotes não coberto pelas exceções será bloqueado por padrão. Em suma, o que não for oficialmente permitido já está expressa e previamente negado.

Negar com DROP ou REJECT?

Ao criar regras de negação utilizando a ação DROP o sistema rejeita o pacote e simplesmente não responde a conexão, já em regras cuja ação definida é REJECT o sistema rejeitará o pacote e notificará o solicitante sobre a rejeição.

Firewall Linux

Servidor: Security

Definindo Regras:

```
iptables -t filter -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
```

A B C

A → Onde a regra será armazenada, ou seja, checada. Sempre indicamos a tabela e a Chain;

B → A regra em si. As informações que passamos na regra são: Origem e Destino do Pacote, Porta do Serviço, Protocolo, etc;

C → A ação da regra, ou seja, o destino do pacote. Se o pacote será aceito, negado, redirecionado, etc.

Firewall Linux

Servidor: Security

Entendendo a Criação de Regras no Iptables:

```
1# iptables -t filter -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
```

Tudo que entrar no Servidor Security com destino ao Localhost (127.0.0.1) será liberado.

```
2# iptables -t filter -S INPUT
3# iptables -t filter -nL
4# ping 127.0.0.1
5# ifconfig eth1
6# ping 192.168.200.1
7# iptables -t filter -nL --line-numbers
```

-A → (append) – Adiciona uma regra do Final da lista de Regras já criadas;

-I → (insert) – Adiciona uma regra no começo da lista de Regras já criadas;

-d → (destination) – Especifica o destino do Pacote.

-s → (source) – Especifica a origem do Pacote.

Tabela de Parâmetros do Iptables

Parâmetros para o iptables		Descrição
-P	--policy	Estabelece a política de acesso a uma chain
-t	--table	Seleciona tabela
-A	--append	Adiciona como última regra da sequência de uma chain
-I	--insert	Insere como primeira regra da sequência de uma chain
-N	--new-chain	Cria uma nova chain
-D	--delete	Remove uma regra
-X	--delete-chain	Elimina todas as regras presentes em chains de usuários
-F	--flush	Elimina todas as regras presentes em uma chain padrão (INPUT,FORWARD etc) ou tablea (Para todas as chains)

Firewall Linux

Servidor: Security

Sobrevivendo no Iptables:

```
1# iptables -t filter -nL --line-numbers
2# iptables -t filter -D INPUT 1
3# iptables -t filter -nL
4# iptables -t filter -A INPUT -j ACCEPT
5# iptables -t filter -nL
6# iptables -t filter -F
7# iptables -t filter -nL
8# iptables -t filter -A INPUT -j ACCEPT
```

-D → (delete) – Deleta uma regra de uma Chain;

-F → (flush) – Deleta todas as regras das Chains de uma tabela;

Tabela de Parâmetros do Iptables

Parâmetros para o iptables		Descrição
-s	--source	Determina a origem do pacote
-d	--destination	Determina o destino do pacote
--dport	--destination-port	Determina a porta de destino
--sport	--source-port	Determina a porta de origem
-i	--in-interface	Define a interface de entrada (input)
-o	--out-interface	Define a interface de saída (output)
-p	--protocol	Seleciona o protocolo (tcp, udp, icmp etc)

Firewall Linux

Servidor: Security

Sobrevivendo no Iptables:

```
1# iptables-save
2# iptables-save > /root/firewall
3# iptables -t filter -F
4# iptables -t filter -nL
5# cat /root/firewall
6# iptables-restore /root/firewall
7# iptables -t filter -nL
8# iptables -t filter -F
```

Todas as regras de firewall são criadas na memória, portanto uma vez que a máquina reinicie tudo é perdido. O iptables não tem um arquivo de configuração, ele oferece dois comandos para que você posso salvar as regras e ativar na inicialização:

iptables-save → Salva as regras num arquivo.

iptables-restore → Ativa as regras armazenadas pelo iptables-save.

Tabela de Alvos do Iptables

Alvo	Descrição do alvo
ACCEPT	O pacote é aceito
REJECT	O pacote é rejeitado imediatamente
DROP	O pacote é nagado silenciosamente (Mais interessante, pois diminui a eficiência de um ataque DOS/DDOS, ou seja, o host de origem fica sem resposta até que a conexão caia).
LOG	Permite que qualquer pacote que se encaixa na regra seja enviado para o syslog ou dmesg

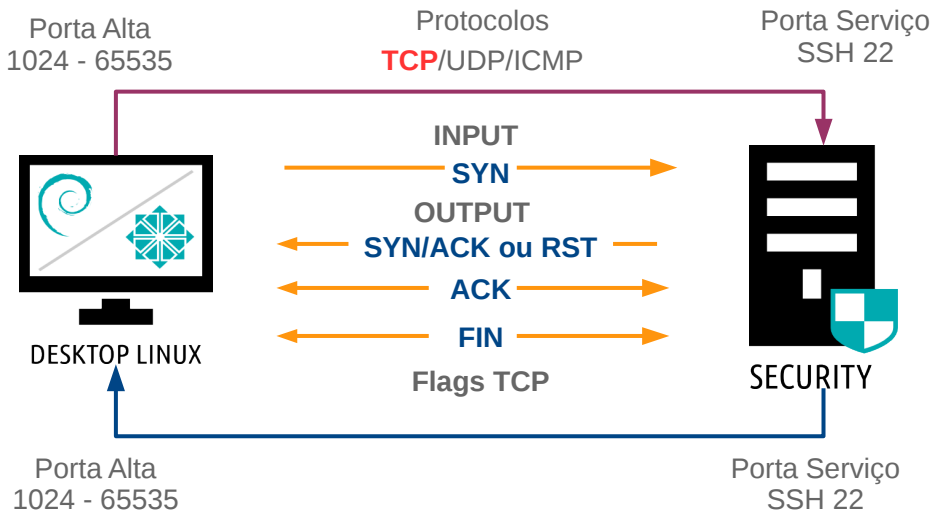
Construindo um Firewall:

Antes de iniciar a construção de um Firewall é importante dominar alguns conceitos de Rede:

- Como funciona uma conexão baseada em cliente e servidor;
- Portas e protocolos dos principais serviços da rede.

Servidor: Security

Como Funciona uma Conexão Baseada em Cliente e Servidor?



Firewall Linux

Servidor: Security

SSH Ativo:

```
#  tcpdump -n -i eth0 host 192.168.200.35 and port 22
IP 192.168.200.35.62939 > 192.168.200.1.22: Flags [S], seq
1470916950, win 65535
IP 192.168.200.1.22 > 192.168.200.35.62939: Flags [S.],
seq 1431520632, ack 1470916951, win 14480
IP 192.168.200.35.62939 > 192.168.200.1.22: Flags [.], ack
1, win 65535
IP 192.168.200.35.62939 > 192.168.200.1.22: Flags [F.],
seq 1809, ack 1512, win 65535
```

Firewall Linux

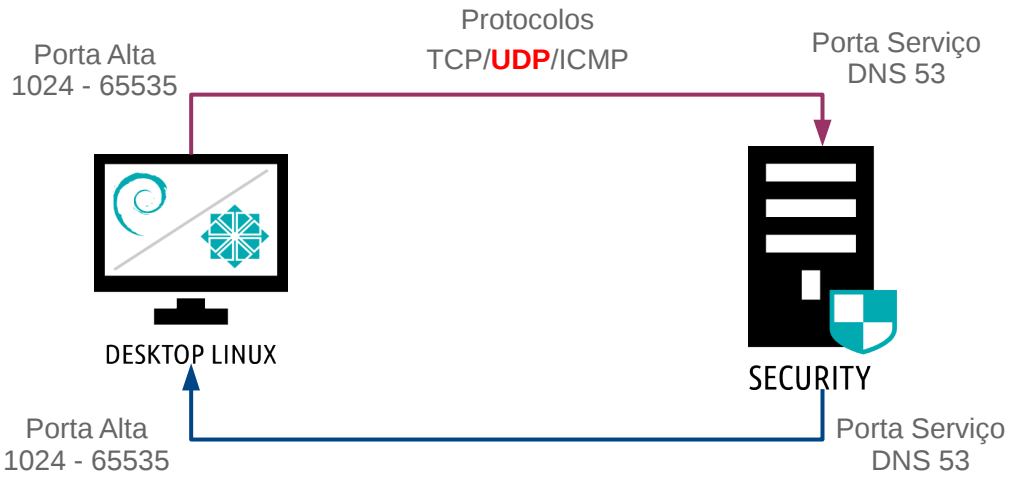
Servidor: Security

SSH Desligado:

```
#  tcpdump -n -i eth0 host 192.168.200.35 and port 22
IP 192.168.100.35.62946 > 192.168.100.1.22: Flags [S], seq
2848827145, win 65535, options
IP 192.168.100.1.22 > 192.168.100.35.62946: Flags [R.],
seq 0, ack 2848827146, win 0, length 0
```

Servidor: Security

Como Funciona uma Conexão Baseada em Cliente e Servidor?



Firewall Linux

Servidor: Security

DNS Ativo:

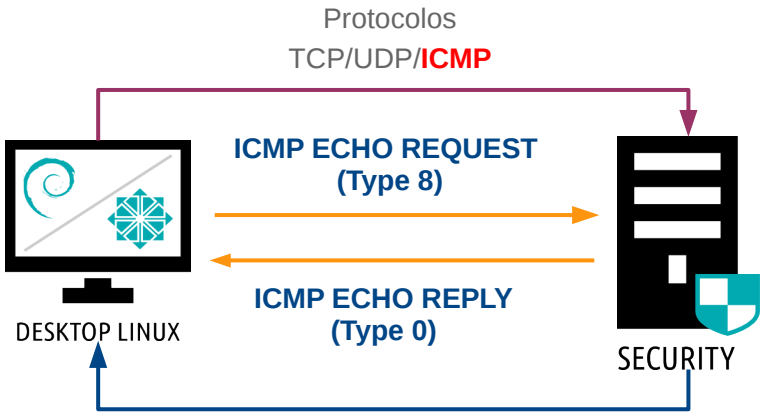
```
#  tcpdump -n -i eth0   port 53
IP 192.168.100.35.57067 > 201.6.2.140.53: 61682+ A?
google.com. (28)
IP 201.6.2.140.53 > 192.168.100.35.57067: 61682 11/4/4 A
74.125.234.68, A 74.125.234.72, A 74.125.234.78, A
74.125.234.73, A 74.125.234.66, A 74.125.234.67, A
74.125.234.64, A 74.125.234.71, A 74.125.234.65, A
74.125.234.70, A 74.125.234.69 (340)
```

DNS Desligado:

```
#  tcpdump -n -i eth0   port 53
IP 192.168.100.1.47619 > 66.118.142.41.53: 65307+ A?
google.com. (28)
IP 66.118.142.41.53 > 192.168.100.1.47619: 65307 Refused-
0/0/0 (28)
```

Servidor: Security

Como Funciona uma Conexão Baseada em Cliente e Servidor?



Ping 4.2.2.2:

```
# tcpdump -n -i eth0   icmp
16:01:44.190864 IP 192.168.100.18 > 4.2.2.2: ICMP echo
request, id 2721, seq 1, length 64
16:01:44.334131 IP 4.2.2.2 > 192.168.100.18: ICMP echo
reply, id 2721, seq 1, length 64
```

Firewall Linux

Servidor: Security

Portas e Protocolos dos Principais Serviços da Rede:

```
1# cat /etc/services
```

SSH	22/tcp	POP3	110/tcp
HTTP	80/tcp	POP3S	995/tcp
HTTPS	443/tcp	IMAP	143/tcp
DNS	53/udp	IMAPs	993/tcp
SMTP	25/tcp	LDAP	389/tcp
SMTPS	465/tcp	OPENVPN	1194/udp
FTP	21/tcp		

Firewall Linux

Servidor: Security

Construindo um Firewall:

1ª → Definir a política das chains da tabela Filter:

```
1# iptables -t filter -P INPUT DROP
2# iptables -t filter -P OUTPUT DROP
3# iptables -t filter -P FORWARD DROP
4# iptables -t filter -nL
5# ping 4.2.2.2
6# ssh 127.0.0.1
```

Firewall Linux

Servidor: Security

2ª → Liberar acesso ao loopback – 127.0.0.1:

```
1# iptables -t filter -A INPUT -s 0/0 -d 127.0.0.1 -j ACCEPT
```

Tudo que entrar no Servidor Security com destino ao Localhost (127.0.0.1) será liberado.

```
2# iptables -t filter -A OUTPUT -s 0/0 -d 127.0.0.1 -j ACCEPT
```

Tudo que sair do Servidor Security com destino ao Localhost (127.0.0.1) será liberado.

```
3# iptables -t filter -nL --line-numbers
4# ssh 127.0.0.1
5# ping 127.0.0.1
```

Firewall Linux

Servidor: Security

3ª → Liberar ping do Firewall para internet e SubRede 1:

```
1# iptables -t filter -A OUTPUT -p icmp -s 0/0 -d 0/0 -j ACCEPT
```

Tudo que sair do Servidor Security sendo protocolo ICMP (ping) com destino a qualquer lugar será liberado.

```
2# iptables -t filter -A INPUT -p icmp --icmp-type 0 -s 0/0 -d 0/0
-j ACCEPT
```

Tudo que entrar no Servidor Security sendo protocolo ICMP do Tipo 0 (echo reply) vindo de qualquer lugar será liberado.

```
3# iptables -t filter -A INPUT -p icmp --icmp-type 8 -s
192.168.200.0/24 -d 0/0 -j ACCEPT
```

Tudo que entrar no Servidor Security sendo protocolo ICMP do Tipo 8 (echo request) vindo da rede 192.168.200.0/24 será liberado.

Firewall Linux

Servidor: Security

3ª → Liberar ping do Firewall para internet e SubRede 2:

```
4# iptables -t filter -A INPUT -p icmp --icmp-type 8 -s
192.168.200.128/25 -d 0/0 -j ACCEPT
```

Tudo que entrar no Servidor Security sendo protocolo ICMP do Tipo 8 (echo request) vindo da rede 192.168.200.128/25 será liberado.

Firewall Linux

Servidor: Security

3ª → Liberar ping do Firewall para internet e rede LAN:

```
1# iptables -t filter -nL --line-numbers
2# ping 8.8.8.8
```

Ping OK, o servidor Firewall está pingando para Internet.

Tente da máquina física pingar o Firewall – Não será permitido!

Agora tente da máquina DataCenter pingar o Firewall.

Ping OK, o servidor DataCenter está pingando o Firewall e vice-versa.

Volte para o servidor Firewall e tente pingar o Google.com.br?

```
3# ping google.com.br
```

Firewall Linux

Servidor: Security

4ª → Liberar consulta DNS a partir do Firewall:

```
1# iptables -t filter -A OUTPUT -p udp -s 200.100.50.99  -d 0/0
--dport 53 -j ACCEPT
```

Tudo que sair do Servidor Security sendo protocolo UDP com destino a qualquer lugar na porta 53 (DNS) será liberado.

```
2# iptables -t filter -A INPUT -p udp -s 0/0 --sport 53 -d
200.100.50.99 -j ACCEPT
```

Tudo que entrar sendo protocolo UDP vindo de qualquer lugar pela porta 53 no Servidor Security será liberado.

```
3# iptables -t filter -nL --line-numbers
4# ping google.com
```

Firewall Linux

Servidor: Security

5ª → Permitir acesso a internet pelo Firewall:

```
1# iptables -t filter -A OUTPUT -p tcp -m multiport -s
200.100.50.99 -d 0/0 --dport 80,443 -j ACCEPT
```

Tudo que sair do Servidor Security sendo protocolo TCP com destino a qualquer lugar nas portas 80 e 443 (http e https) será liberado.

```
2# iptables -t filter -A INPUT -p tcp -m multiport -s 0/0
--sport 80,443 -d 200.100.50.99 -j ACCEPT
```

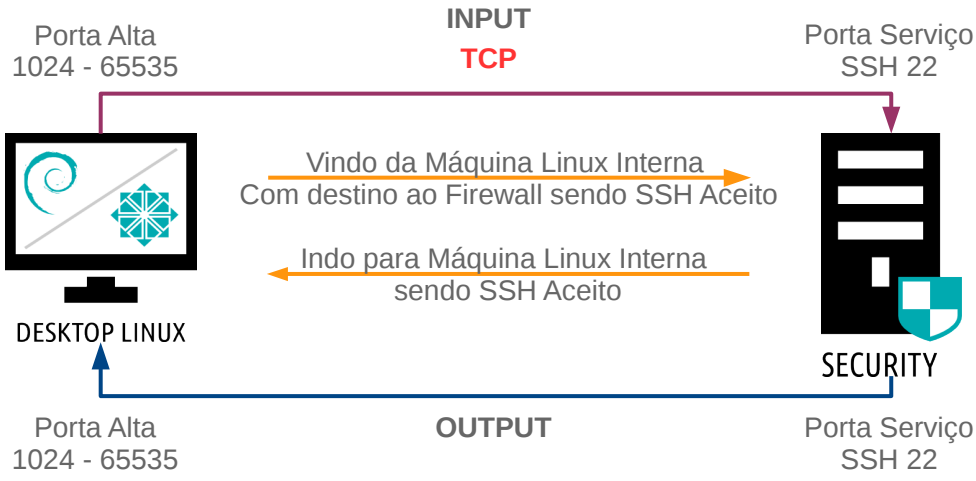
Tudo que entrar sendo protocolo TCP vindo de qualquer lugar pelas portas 80 e 443 no Servidor Security será liberado.

```
3# iptables -t filter -nL --line-numbers
4# apt-get update
```

Laboratório Dexter

Servidor: Security

6ª → Libere acesso SSH da máquina Linux Interna para o Firewall:



Laboratório Dexter

Servidor: Security

6ª → Libere acesso SSH da máquina Linux Interna para o Firewall:

```
1# iptables -t filter -A INPUT -p tcp -s 192.168.200.10 -d
200.100.50.99 --dport 22 -j ACCEPT
```

Tudo que entrar vindo da Máquina Linux Interna sendo protocolo TCP com destino ao Servidor Security na porta 22 (SSH) será liberado.

```
2# iptables -t filter -A OUTPUT -p tcp -s 200.100.50.99 --sport
22 -d 192.168.200.10 -j ACCEPT
```

Tudo que sair sendo protocolo TCP saindo do Servidor Security na porta 22 com destino a Máquina Linux Interna será liberado.

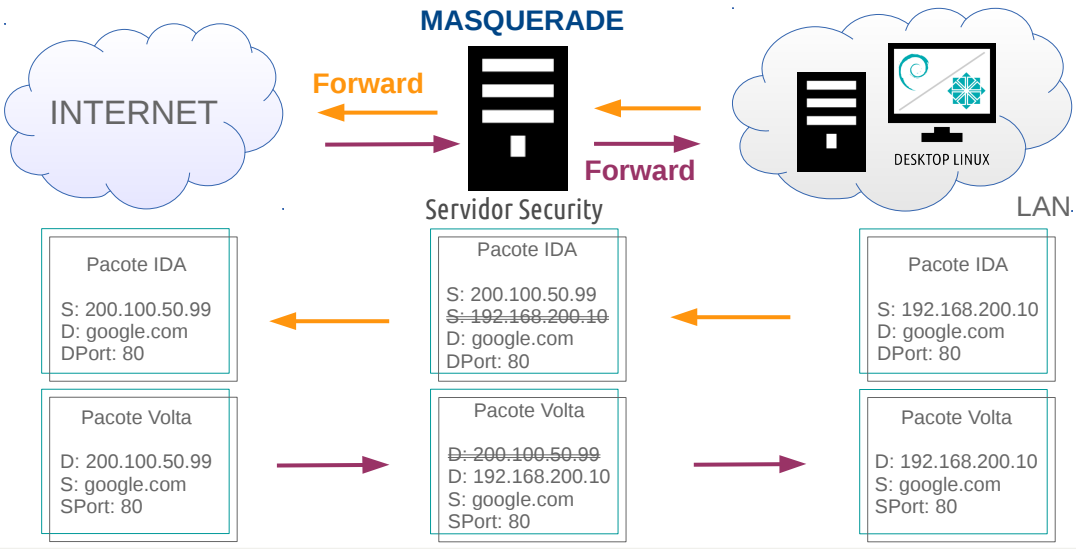
```
3# iptables -t filter -nL --line-numbers
```

Tente da Máquina Linux Interna acessar o Firewall por SSH – Será permitido!

Firewall Linux

Servidor: Security

Como Funciona o Compartilhamento de Internet?



Firewall Linux

Servidor: Security

7ª → Libere acesso a internet para as máquinas das SubRede 1 e 2:

Para trabalhar com MASQUERADE, NAT ou até mesmo roteamento de pacotes por tabela de roteamento é necessário ativar o repasse de pacotes entre placas fisicamente no kernel:

```
1# vim /etc/sysctl.conf
   28 net.ipv4.ip_forward=1
2# sysctl -p
3# cat /proc/sys/net/ipv4/ip_forward
```

Firewall Linux

Servidor: Security

7ª → Libere acesso a internet para as máquinas da SubRede 1:

```
1# iptables -t nat -A POSTROUTING -s 192.168.200.0/24  -d 0/0
-j MASQUERADE
```

Tudo que vier da Rede Interna com destino a Internet será MASCARADO

```
2# iptables -t filter -A FORWARD -p tcp -m multiport -s
192.168.200.0/24 -d 0/0 --dport 80,443 -j ACCEPT
```

Tudo que vier da Rede Interna com destino a Internet nas portas 80 443 eu permito o repasse de pacotes

```
3# iptables -t filter -A FORWARD -p tcp -m multiport -s 0/0
--sport 80,443 -d 192.168.200.0/24  -j ACCEPT
```

Tudo que vier da Internet nas portas 80 e 443 com destino a Rede Interna eu permito o repasse de pacotes

Firewall Linux

Servidor: Security

7ª → Libere acesso a internet para as máquinas da SubRede 2:

```
1# iptables -t nat -A POSTROUTING -s 192.168.200.128/25  -d 0/0
-j MASQUERADE
```

Tudo que vier da Rede Interna com destino a Internet será MASCARADO

```
2# iptables -t filter -A FORWARD -p tcp -m multiport -s
192.168.200.128/25 -d 0/0 --dport 80,443 -j ACCEPT
```

Tudo que vier da Rede Interna com destino a Internet nas portas 80 443 eu permito o repasse de pacotes

```
3# iptables -t filter -A FORWARD -p tcp -m multiport -s 0/0
--sport 80,443 -d 192.168.200.128/25  -j ACCEPT
```

Tudo que vier da Internet nas portas 80 e 443 com destino a Rede Interna eu permito o repasse de pacotes

Firewall Linux

Servidor: Security

8ª → Libere Acesso a Consulta DNS para as Máquinas da SubRede 2:

```
1# iptables -t filter -A FORWARD -p udp -s 192.168.200.128/25
-d 0/0 --dport 53 -j ACCEPT
```

Tudo que vier da rede interna com destino a internet na porta 53 eu permito o repasse de pacotes

```
2# iptables -t filter -A FORWARD -p udp -s 0/0 --sport 53 -d
192.168.200.128/25 -j ACCEPT
```

Tudo que vier da internet na porta 53 com destino a rede interna eu permito o repasse de pacotes

Tente do Servidor Datacenter acessar a Internet – Será permitido!

Firewall Linux

Servidor: Security

8ª → Libere Acesso a Consulta DNS para as Máquinas da SubRede 1:

```
1# iptables -t filter -A FORWARD -p udp -s 192.168.200.0/24 -d
0/0 --dport 53 -j ACCEPT
```

Tudo que vier da rede interna com destino a internet na porta 53 eu permito o repasse de pacotes

```
2# iptables -t filter -A FORWARD -p udp -s 0/0 --sport 53 -d
192.168.200.0/24 -j ACCEPT
```

Tudo que vier da internet na porta 53 com destino a rede interna eu permito o repasse de pacotes

Tente do Servidor Audit acessar a Internet – Será permitido!

Firewall Linux

Servidor: Security

8ª → Libere Acesso a Consulta DNS para as Máquinas da SubRede 2:

```
1# iptables -t filter -A FORWARD -p udp -s 192.168.200.128/25
-d 0/0 --dport 53 -j ACCEPT
```

Tudo que vier da rede interna com destino a internet na porta 53 eu permito o repasse de pacotes

```
2# iptables -t filter -A FORWARD -p udp -s 0/0 --sport 53 -d
192.168.200.128/25 -j ACCEPT
```

Tudo que vier da internet na porta 53 com destino a rede interna eu permito o repasse de pacotes

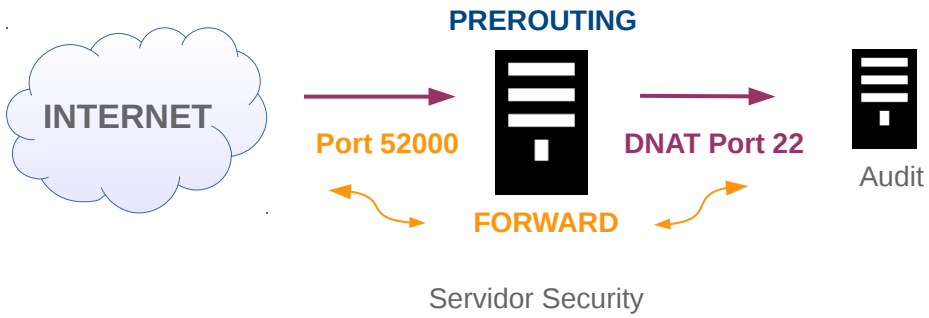
Tente do Servidor Datacenter acessar a Internet – Será permitido!

Firewall Linux

9ª → Redirecione o serviço SSH para os servidores internos:

Porta 52000	→	Servidor Audit	→ Porta 22
Porta 53000	→	Servidor Storage	→ Porta 22
Porta 54000	→	Servidor FileServer	→ Porta 22
Porta 55000	→	Servidor WebServerInterno	→ Porta 22
Porta 56000	→	Servidor Datacenter	→ Porta 22

Como Funciona o Redirecionamento de Portas?



Servidor Security

Firewall Linux

Servidor: Security

9ª → Redirecione o serviço SSH para os servidores internos:

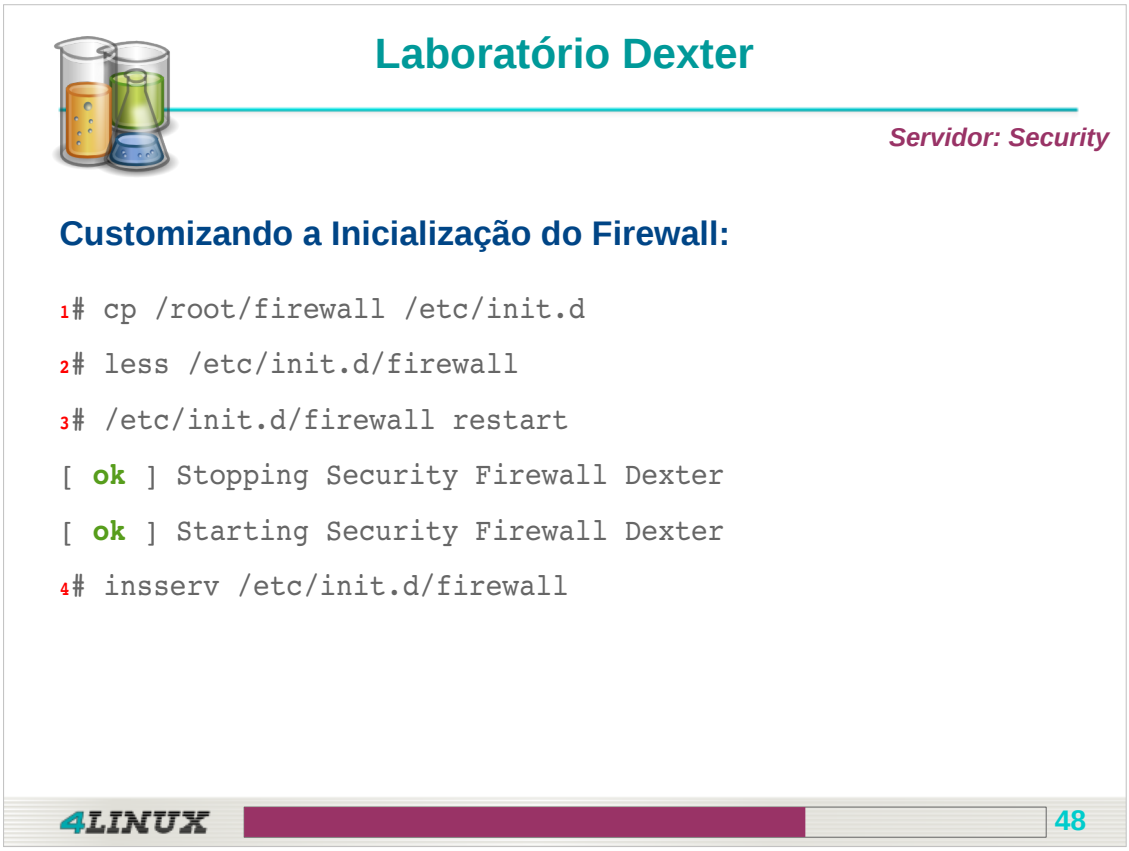
```
1# iptables -t nat -A PREROUTING -p tcp -s 0/0 -d 200.100.50.99
--dport 52000 -j DNAT --to 192.168.200.30:22
```

Tudo que vier da Internet com destino ao Servidor Security na porta 52000 será redirecionado ao Servidor Audit na porta 22

```
2# iptables -t filter -A FORWARD -p tcp -s 0/0 -d
192.168.200.30 --dport 22 -j ACCEPT
```

Tudo que vier da Internet com destino ao Audit na porta 22 eu permito o repasse de pacotes

```
3# iptables -t filter -A FORWARD -p tcp -s 192.168.200.30
--sport 22 -d 0/0 -j ACCEPT
```



Customizando a Inicialização do Firewall

O "script" acima foi adicionado aos níveis de execução do sistema, para ser carregado sempre que a máquina for ligada.

Um bom Firewall é aquele que bloqueia tudo e libera somente o necessário por isso as primeiras providencias a tomar é bloquear tudo.

A POLICE (determinada com o parâmetro -P) é a regra default para qualquer tipo de comunicação, portanto se não existir uma regra especifica para a passagem de pacotes a POLICE irá bloquear a passagem.

Resumão iptables (parte 1/2)

Descrição	Parâmetros Abreviado	Parâmetros Completo
Estabelece a politica de acesso de uma chain	`-P`	`--policy`
Selecionar a Tabela	`-t`	`--table`
Adiciona como última regra da sequência de uma Chain	`-A`	`--append`
Insere como primeira regra da sequência de uma Chain	`-I`	`--insert`
Remove uma Regra	`-D`	`--delete`
Elimina todas as regras presentes em uma Chain padrão (INPUT,FORWARD etc) ou tabela (Para todas as Chains).	`-F`	`--flush`

Resumão iptables (parte 2/2)

Descrição	Parâmetros Abreviado	Parâmetros Completo
Determina a origem do Pacote	`-s`	`--source`
Determina o Destino do Pacote	`-d`	`--destination`
Define a porta de Destino	`--dport`	`--destination-port`
Define a porta de Origem	`--sport`	`--source-port`
Seleciona o Protocolo (tcp,udp,icmp etc)	`-p`	`--protocol`

Dica

Servidor: Security

Colocando Comentário no iptables para Agilizar na Leitura da Regra:

```
1# iptables -t filter -A INPUT -s 0/0 -d 127.0.0.1 -j ACCEPT
-m comment --comment "Aceita tudo Loopback"
```

Tudo que vier de qualquer lugar com destino a Loopback (127.0.0.1) permiti o acesso.

```
2# iptables -nL
```

Facilita para obter de forma rápida o objetivo daquela regra, sem a necessidade de entrar em arquivos.

Dica

Servidor: Security

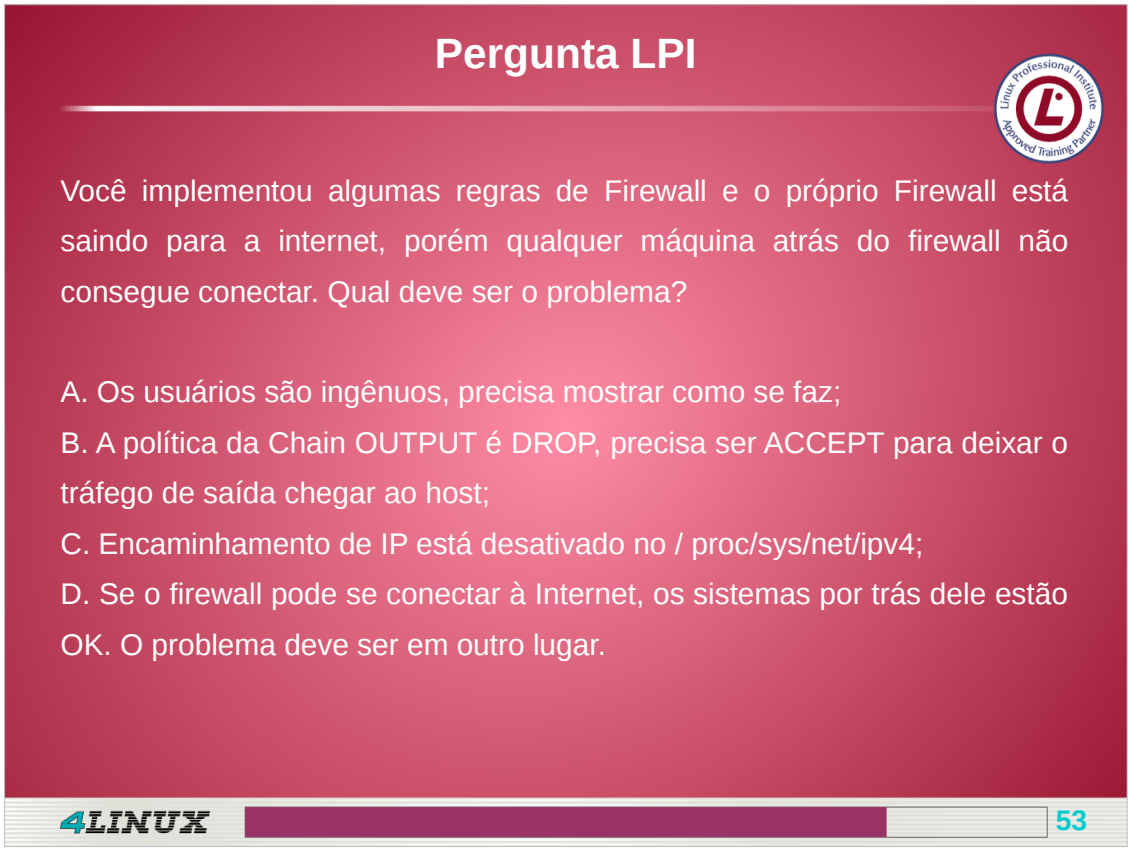
Criando Registro de LOG do iptables

```
1# iptables -t filter -A INPUT -s 0/0 -d 200.100.50.99 -p tcp
  --dport 22 -j LOG --log-prefix '[Acesso SSH Security]'
```

Tudo que vier de qualquer lugar com destino a 200.100.50.99 na porta 22 cria o log "Acesso SSH Security".

```
2# iptables -nL
3# tail -f /var/log/messages
```

Tente da máquina física acessar o Firewall por SSH!

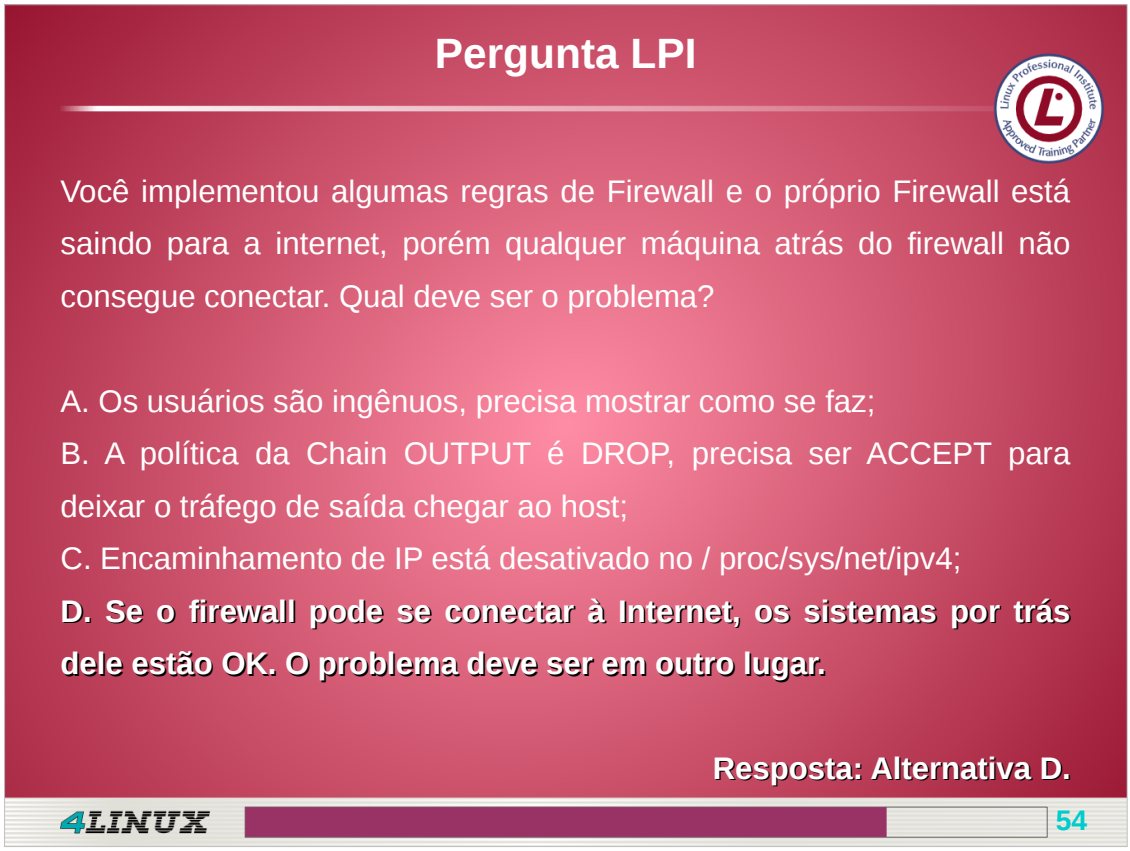


Alternativa D: Resposta Correta!

Lembre-se: Para que um sistema Linux possa “repassar” pacotes é necessário que o suporte a encaminhamento esteja habilitado no kernel, essa opção deverá ser encontrada no arquivo /etc/sysctl.conf ou diretamente no /proc.

Na 4Linux temos dois treinamentos de segurança da informação: PenTest, testes de vulnerabilidades de redes e Segurança de servidores Linux usando a ISO27002.

A Prova do LPI número 303 é focada em Segurança. Conceitos e comando básicos do "iptables"são cobrados na prova 202.

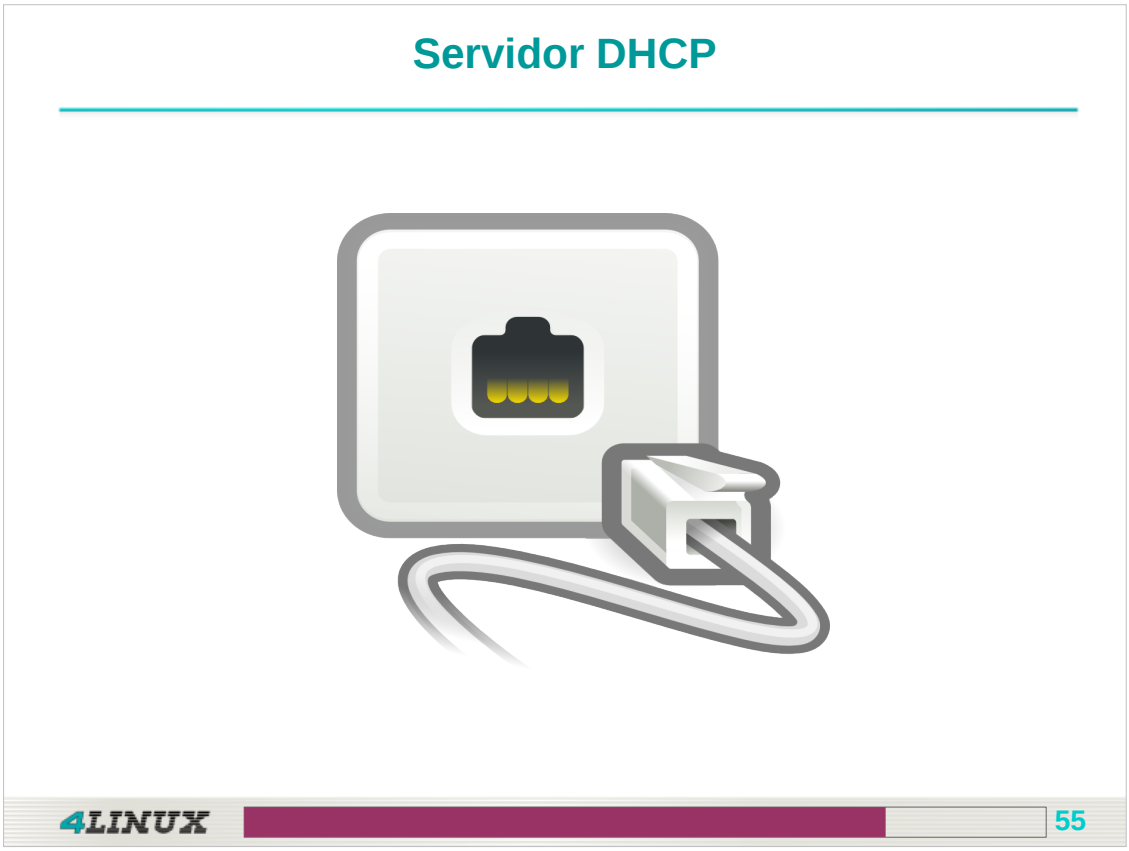


Alternativa D: Resposta Correta!

Lembre-se: Para que um sistema Linux possa “repassar” pacotes é necessário que o suporte a encaminhamento esteja habilitado no kernel, essa opção deverá ser encontrada no arquivo /etc/sysctl.conf ou diretamente no /proc.

Na 4Linux temos dois treinamentos de segurança da informação: PenTest, testes de vulnerabilidades de redes e Segurança de servidores Linux usando a ISO27002.

A Prova do LPI número 303 é focada em Segurança. Conceitos e comando básicos do "iptables"são cobrados na prova 202.



Cenário

A empresa Dexter Courier precisa de informações precisas sobre suas máquinas no que diz respeito à origem dos acessos aos servidores, portanto ter todos os computadores da rede mapeados com endereço IP fixo.

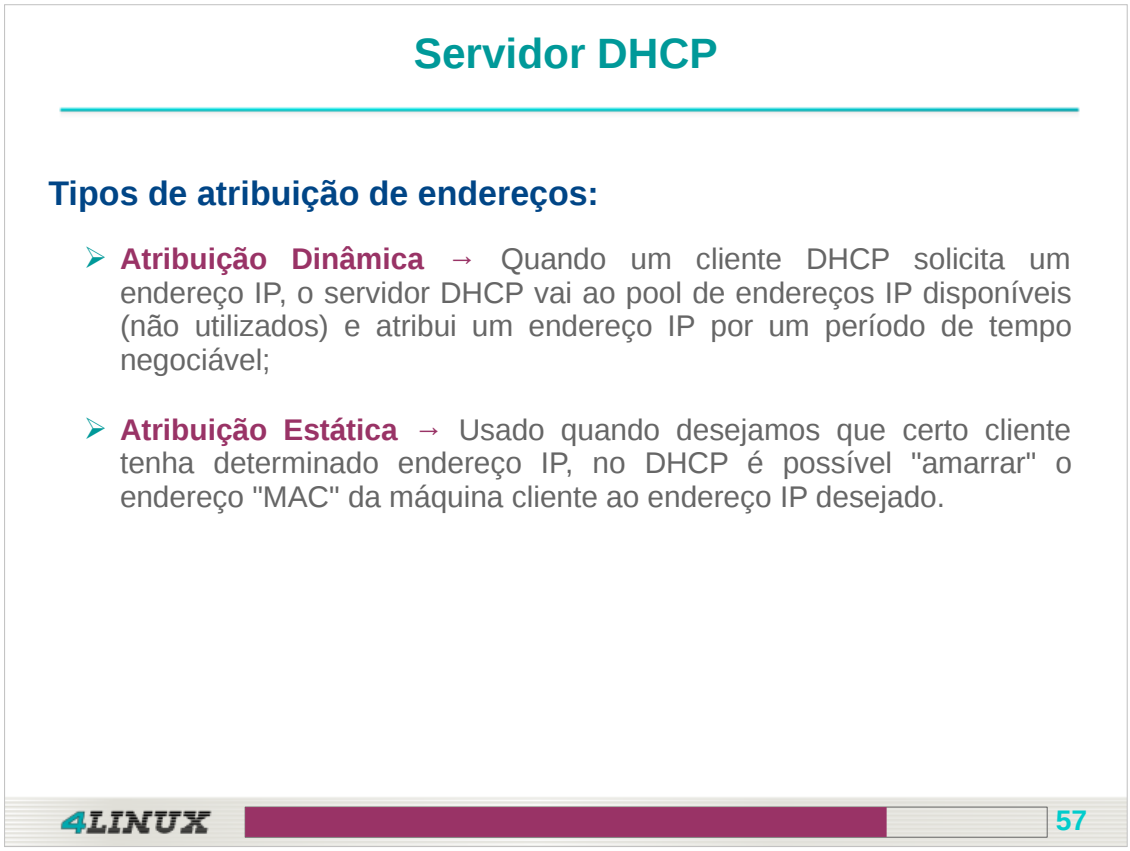
Proposta de solução

Uma grande vantagem que podemos encontrar no DHCP é o de “amarrar” o endereço MAC das máquinas a um determinado IP, fazendo com que somente quem estiver com o MAC ADDRESS cadastrado poderá utilizar a internet e os demais serão bloqueados. Além disto, teremos um controle melhor sobre a origem dos acessos aos serviços da empresa.

Fundamentação

Criado e mantido pelo ISC (Internet Systems Consortium), um grupo sem fins lucrativos dedicado a desenvolver serviços de infra-estrutura usados na Internet, incluindo o Bind e o NTPD. Caso esteja curioso, a página com o código fonte é a: http://www.isc.org/sw/dhcp/.

O protocolo DHCP (Dynamic Host Configuration Protocol) funciona nas camadas 2 e 3 do modelo OSI e é amplamente utilizado para oferecer endereço IP a um "host" que ainda não está configurado, o que oferece uma flexibilidade ao Administrador de Redes.



Fundamentação

O que um servidor DHCP faz é bastante simples. Ele responde aos pacotes de broadcast das estações, enviando um pacote com um dos endereços IP disponíveis e os demais dados da rede.

Os pacotes de broadcast são endereçados ao endereço "255.255.255.255" e retransmitidos pelo switch da rede para todas as portas, diferente dos pacotes endereçados a um endereço específico, que são transmitidos apenas na porta relacionada a ele.

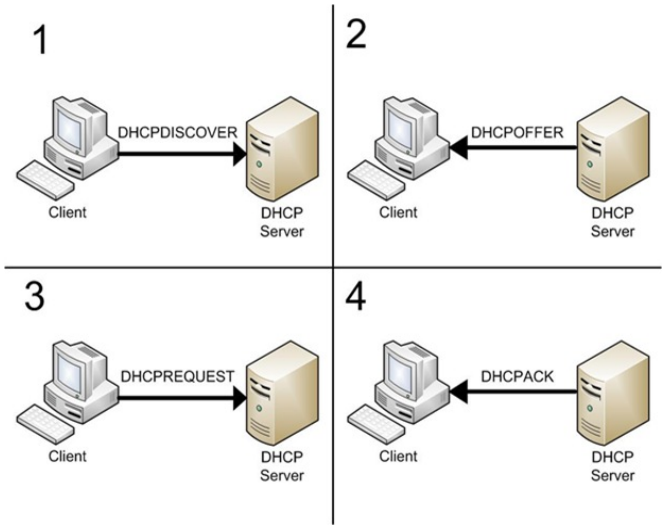
Periodicamente o servidor DHCP verifica se as estações ainda estão lá, exigindo uma renovação do "aluguel" do endereço IP. Isso permite que os endereços IP sejam gastos apenas com quem realmente estiver online, evitando que os endereços disponíveis se esgotem.

Servidor DHCP

Servidor: Linux Interna

Cliente DHCP:

```
1# dhclient -r
2# dhclient -v eth0
3# ifconfig eth0
```



Atenção: Antes de desligar a **DesktopLinux**, altere a configuração de rede.

4LINUX 58

DHCPDISCOVER → Um cliente envia um quadro "broadcast"/ (destinado a todas as máquinas) com um pedido "DHCP";

DHCPOFFER → O servidor "DHCP" captura o quadro e oferece um Endereço IP ao "host" requisitante;

DHCPREQUEST → O cliente envia um "DHCP REQUEST" endereçado para o servidor "DHCP" aceitando o IP;

DHCPACK → Esse é o pacote que confirma a atribuição de uma configuração de rede a um cliente, ou seja, aquele cliente agora possui configurações distribuídas pelo servidor "DHCP";

Servidor DHCP

Servidor: Audit

Servidor DHCP:

```
1# apt-get install isc-dhcp-server
2# cd /etc/dhcp/
3# mv dhcpd.conf dhcpd.conf.dist
4# vim dhcpd.conf.dist
5# vim dhcpd.conf
```

4LINUX 59

DHCPNAK → Caso o cliente não aceite aquele endereço IP, ele enviará um "DHCP-NAK" para o servidor, e realizará o "DHCPDISCOVER" novamente.

DHCPDECLINE → Caso o IP já esteja sendo utilizado por outro Host, o cliente envia um DHCPDECLINE anunciando ao servidor que o endereço já é utilizado;

DHCPRELEASE → Se por algum motivo o cliente deixa de usar o endereço IP, o mesmo manda um DHCPRELEASE dizendo que aquele IP ja pode ser usado por outro Host;

DHCPINFORM → Utilizado quando o Cliente já possui endereço IP, mas precisa de outras informações, como o servidor DNS por exemplo, o mesmo envia um DHCPINFORM pedindo tais informações para o servidor DHCP.

Servidor DHCP

Servidor: Audit

```
ddns-update-style none;
log-facility local7;
subnet 192.168.200.0 netmask 255.255.255.0 {
range 192.168.200.100 192.168.200.110;
server-identifier audit;
option domain-name "dexter.com.br";
option domain-name-servers 192.168.200.30,192.168.200.130;
option routers 192.168.200.1;
default-lease-time 600;
max-lease-time 7200;
min-lease-time 1000;
}
```

4LINUX 60

• **ddns-update-style** → Esquema de armazenamento das informações de redes dos clientes. Atualmente são implementados dois esquemas de : O modo de atualização do DNS improvisado (ad-hoc) e o modo de atualização do esquema de interação do intervalo DHCP-DNS (interim);

• **deny unknown-clients** → Nega acesso a MAC não cadastrado;

• **log-facility local7** → Criaremos uma entrada no /etc/rsyslog.conf para os logs do servidor DHCP;

• **subnet 192.168.1XX.0 netmask 255.255.255.0** → Qual rede o DHCP irá responder quando for solicitado;

• **range 192.168.1XX.100 192.168.1XX.110** → Faixa de IPs que será disponibilizado aos clientes;

Servidor DHCP

Servidor: Audit

Reinicie o Serviço:

```
1# /etc/init.d/isc-dhcp-server stop
2# /etc/init.d/isc-dhcp-server start
3# ps aux | grep dhcp
4# tail -f /var/lib/dhcp/dhcpd.leases
```

Ligue o desktop Linux para testar seu servidor DHCP

```
5# dhclient -v eth0
```

• **option domain-name dexter.com.br** → Nome de domínio do cliente;

• **option domain-name-servers 192.168.1XX.3,192.168.1XX.2** → Essa opção lista os servidores de nomes (DNS) a serem utilizados para resolução de nomes;

• **option routers 192.168.1XX.1** → O cliente, além do número IP, recebe também a informação do número do "gateway" de sua rede;

• **default-lease-time 600** → Servidores "DHCP" cedem endereços sob pedido por um tempo pré-determinado. O padrão nesse exemplo é ceder o endereço IP por 600 segundos, ou 10 minutos;

• **max-lease-time 7200** → Caso o cliente solicite um tempo maior, o tempo máximo permitido será de 7.200 segundos (2 horas);

• **min-lease-time 1000** → Tempo mínimo de concessão de 1000 segundos (Aproximadamente 16 minutos);

Servidor DHCP

Servidor: Audit

DHCP Estático:

```
1# vim /etc/dhcp/dhcpd.conf
host linux-interna {
    hardware ethernet 00:00:00:00:00:00;
    fixed-address 192.168.200.110;
}
2# /etc/init.d/isc-dhcp-server start
```

Faça uma nova solicitação no DesktopLinux:

```
3# dhclient -v
```

Laboratório Dexter

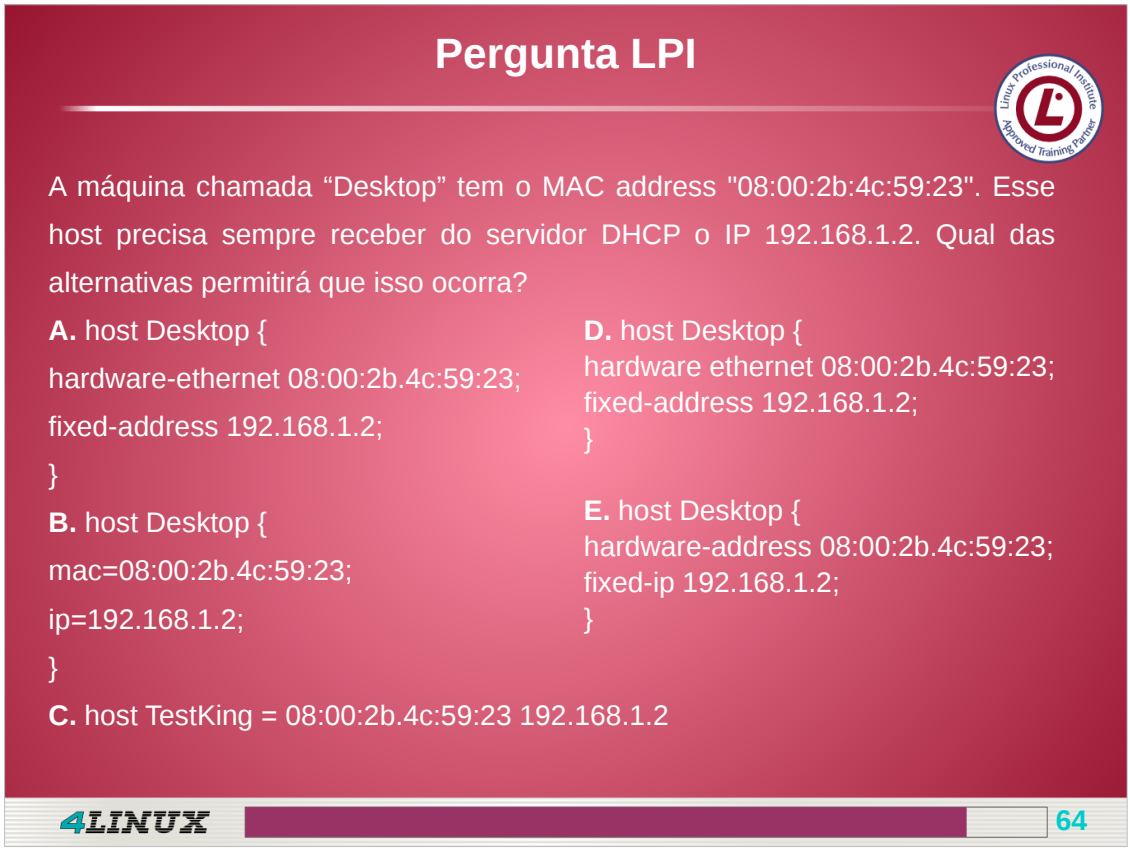
Servidor: Audit

- Ligue a máquina Windows e repita o teste do DHCP Dinâmico:

```
1# tail -f /var/lib/dhcp/dhcpd.leases
```

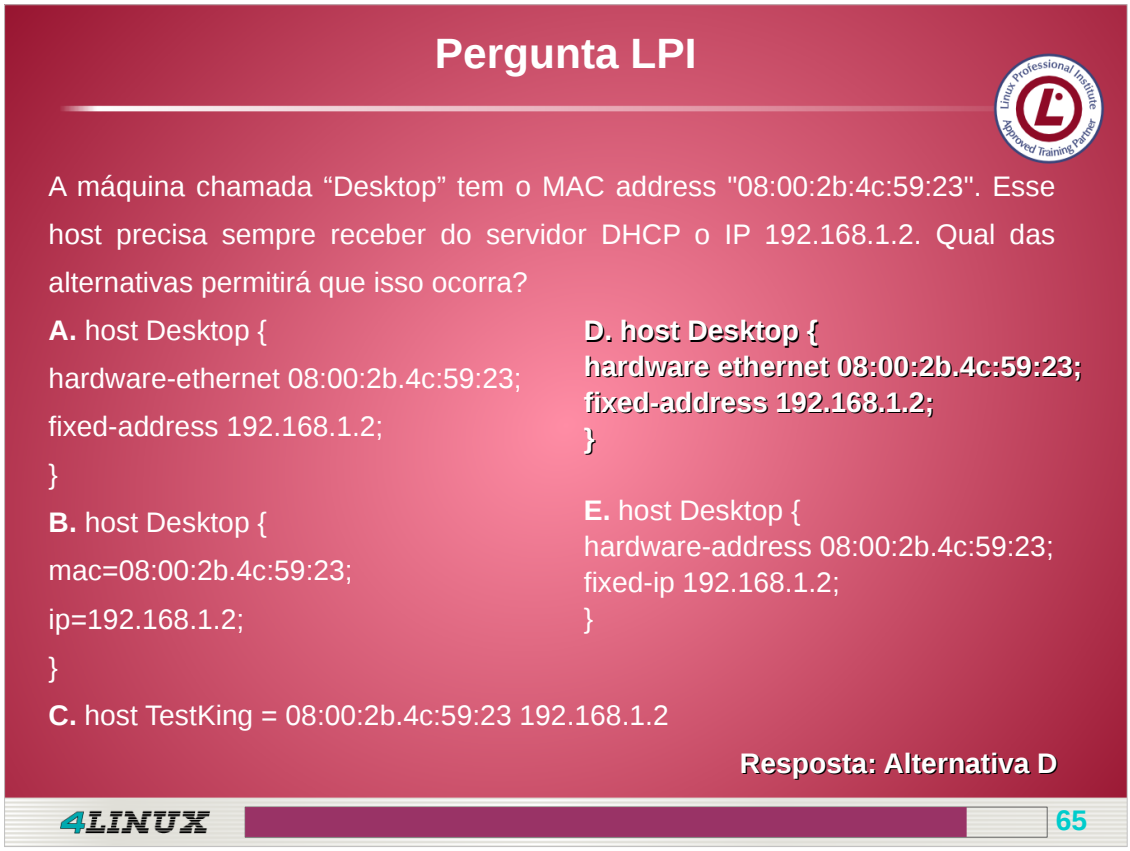
- Em seguida, fixe o IP via MAC Address para a máquina Windows Interna:

```
host win8 {
hardware ethernet 00:00:00:00:00:00;
fixed-address 192.168.200.111;
}
```



Alternativa D: Resposta Correta!

Mutias questões LPI cobram comandos exatos alterando alterando apenas pequenos detalhes como no exemplo acima, por isso ao se preparar para a prova preste atenção na sintaxe de configuração e execução dos comandos apresentados em cada tópico.



Alternativa D: Resposta Correta!

Mutias questões LPI cobram comandos exatos alterando alterando apenas pequenos detalhes como no exemplo acima, por isso ao se preparar para a prova preste atenção na sintaxe de configuração e execução dos comandos apresentados em cada tópico.

Próximos Passos

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- Executar as tarefas do **Practice Lab**;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- Responder as questões do **Teste de Conhecimento** sobre o conteúdo visto em aula.

Mãos à obra!