

Eletrônico



Estratégia
CONCURSOS

Aula

Sistemas Oper. e Arquitetura de Sistemas pl Câmara de Ouro Preto (Analista de Sistemas) - Pós-Edital

Professor: Celson Carlos Martins Junior, Equipe Informática e TI, Evandro Dalla Vecchia Pereira

Esclarecimentos Iniciais	2
1 – Firewall Linux	3
1.1 Gerações de Firewall	4
1.2 Classificação dos firewalls Linux.....	5
1.3 Cadeias e “Chains”	6
1.4 Iptables	7
1.5 Administração Iptables.....	8
1.6 Tabela Filter.....	9
1.7 Tabela NAT	10
1.8 Tabela Mangle.....	12
1.9 Scripts Administração Iptables	12
1.10 Resolução de questões	15



Pessoal, o objetivo desta aula é entendermos os conceitos e noções básicas de administração de Firewall Linux.

Este assunto tem tido presença rara nas últimas provas da banca.

Nessa aula, resolveremos questões recentes da banca sobre o tópico desta aula.

Pessoal, antes de iniciar nosso assunto propriamente dito, precisamos esclarecer alguns pontos.

Nossa abordagem será **descritiva**, ou seja, iremos funcionalidades, características e comandos, sempre recorrendo às questões de concursos para nos balizar.

Além de entender as noções básicas, um dos nossos objetivos é auxiliá-los a identificar o “modus operandi” da banca e verificar quais conceitos são mais abordados.

Atenção, como não há questões suficientes de apenas uma banca para cobrir todos os tópicos previstos no edital, iremos nos valer de questões de diversas bancas.

Para facilitar nossa vida, no decorrer do texto, os conceitos preferidos da banca foram acompanhados com um dos logos do Estratégia abaixo:



TOME NOTA!



Recomendamos que em caso de dúvidas sobre algum ponto não abrangido no curso, recorram as páginas de manual (man pages) do sistema Linux ao a documentação relativa.

Para padronizar nosso entendimento, os comandos serão exibidos no seguinte formato:

```
#
```

Sem mais delongas, vamos a nossa aula, Ok. 😊



Pessoal, um **firewall** é um ponto que interliga duas ou mais redes, públicas (como a Internet) e privadas (redes locais).

O termo original “firewall” ou “parede de fogo” refere-se a uma barreira física usada para impedir que incêndios se alastrassem entre muros em casas ou apartamentos. Analogamente, um firewall no sentido informático impede que comunicações indesejadas ultrapassem o perímetro de uma rede.

Numa grande maioria dos casos, consiste numa configuração de hardware ou software que está na fronteira (perímetro) entre uma rede local e uma rede externa e que controla todo o fluxo de comunicações entre estas redes.

O papel principal do firewall é mediante algum critério definido autorizar algumas requisições deste fluxo de requisições entre duas redes, baseando-se nos endereços de origem e destino de um pacote IP ou por exemplo das portas destino e opções TCP de um pacote.

Segundo a **RFC 2647**, “Benchmarking Terminology for Firewall performance”, um firewall é “um dispositivo, serviço de sistema operativo, ou aplicação que estabelece uma política de controlo de acesso entre redes.”

Mais especificamente, o termo firewall possui um número de diferentes significados consoante os mecanismos utilizados para implementar a firewall, a camada da pilha TCP/IP em que o firewall opera, e as arquiteturas de rede e routing utilizadas.

Um firewall pode ser de vários tipos, firewall estática, dinâmica de filtro-de-pacotes, pode ser um firewall de aplicação ou mesmo um servidor proxy. Pode ainda ser baseado em software ou em hardware.

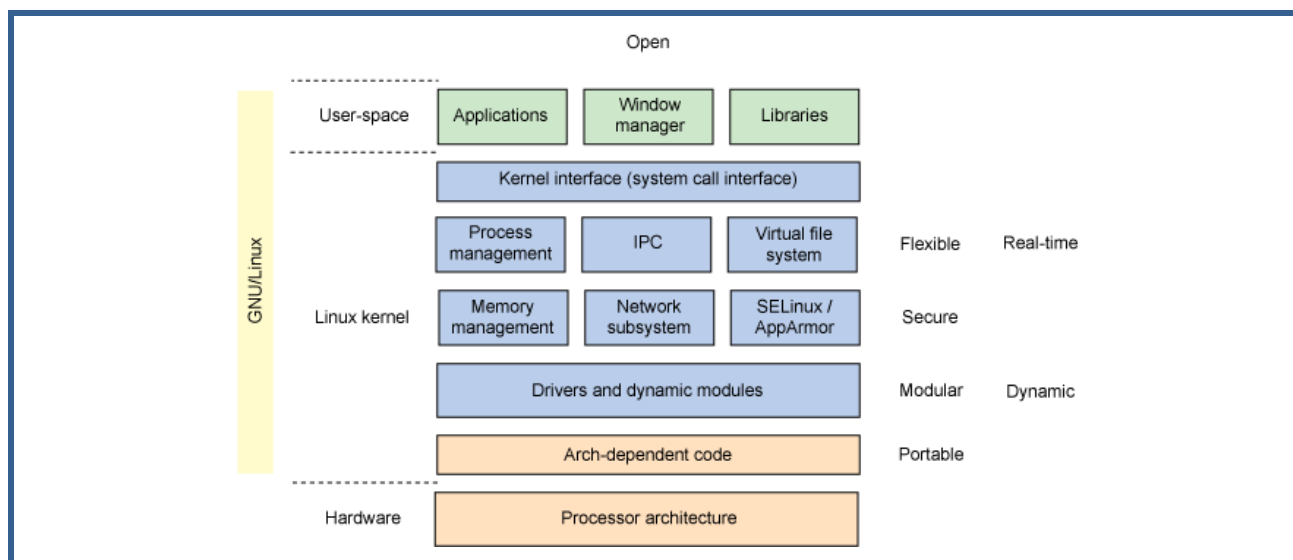
Firewalls podem atuar na segurança da **camada de transporte ou na camada de aplicação** da pilha TCP/IP.

Caso não tenha conhecimentos da pilha TCP/IP, peço que recorra a aula relativa a este tópico antes de prosseguir, visto que é essencial para compreensão.

Firewall da camada de transporte filtram pacotes com base na informação dos cabeçalhos UDP e TCP, como porta origem e destino ou com flags específicas do cabeçalho TCP.



firewall em sentido estrito. Na figura abaixo ilustra-se o SELinux.



1.1 GERAÇÕES DE FIREWALL

A **primeira geração** de firewalls surge em meados de 1985, através de routers IP com regras de filtragem de pacotes que podiam ser modificadas manualmente pelo administrador.

Eram firewalls fáceis de manter porque existiam poucos serviços que disponibilizados para o exterior: acesso seguro a telnet, FTP, E-mail e news. Uma vez que os routers IP deixam passar o tráfego directamente, os utilizadores apenas precisavam ter um IP válido ou acesso á intranet.

A **segunda geração** de firewalls, desenvolvida entre 1989-1990, foram firewalls/gateways ao nível do circuito, impedindo ligações directas entre redes, autorizando-as baseadas no endereço.

Durante a **terceira geração** de firewalls, apareceram as application layer firewalls (firewalls da camada de aplicação), sendo os casos mais especiais bastion hosts a correrem serviços proxy.

A **quarta geração** de firewalls que consistiu no desenvolvimento de firewall dinâmicas de filtro de pacotes que modificavam dinamicamente as regras de filtragem.

A tecnologia de filtragem dinâmica de pacotes e, ao contrário de firewalls anteriores, foi a primeira com instalação e administração simplificada, não sendo necessário editar ficheiros ASCII para configuração do firewall.



arquitetura de proxy a nível do Kernel, que monitora o tráfego em múltiplas camadas da pilha do protocolo TCP/IP.

1.2 CLASSIFICAÇÃO DOS FIREWALLS LINUX

Firewall é um termo vulgarmente atribuído a diferentes sistemas tais como firewalls estáticas de filtro de pacotes ou routers de filtro de pacotes, firewalls dinâmicas ou stateful firewalls, gateways de aplicação, ou a servidores proxy.

No entanto podemos classificá-los nas categorias a seguir:



TOME NOTA!

Firewalls estáticos de filtro-de-pacotes “stateless”

Um firewall estático de filtro de pacotes é normalmente implementado dentro do sistema operativo ou hardware do router e opera na camada de rede (IP network layer), e muitas vezes também na camada de transporte (transport layer).

Protegem o sistema fazendo decisões de routing após a filtragem dos pacotes baseada na informação presente nos cabeçalhos dos pacotes. Estas decisões são tomadas pacote a pacote.

Em sistemas maiores, um filtering router (também designado screening router) é muitas vezes colocado antes do firewall para fazer filtragem inicial com o objectivo de reduzir o esforço de processamento requerido pelo firewall.

Um firewall de **filtro de pacotes utiliza uma listas de regras de aceitação e/ou negação**. Estas regras explicitamente definem que pacotes é que serão admitidos pela interface de rede, e para isso examinam informação presente nos cabeçalhos dos pacotes.

Após esta análise o pacote ou é enviado para o seu destino, ou descartado (rejeitado), ou bloqueado e enviado um pacote com condição de erro para a máquina remetente (negado).

Estas regras são baseadas em diversas informações como a interface de rede e respectivo endereço IP, o endereço IP origem e destino (camada de rede), as portas origem e destino, TCP ou



Firewalls dinâmicos, ou Firewalls de inspeção “stateful”

Os firewalls dinâmicos operam nas camadas de rede e de transporte, podendo filtrar pacotes com base em certas informações de protocolos de acesso ao meio e mesmo de protocolos de camadas superiores à camada de transporte.

Este tipo de firewalls de filtro de pacotes mantêm informação sobre sessões TCP ou troca de pacotes UDP. Os pacotes são filtrados no contexto de uma sessão e não isoladamente.

Por exemplo, este tipo de firewalls sabe que um pacote TCP com a flag ACK deve ser descartado caso não tenha recebido um pacote inicial com a flag SYN. Sabe se um determinado pacote UDP é resposta esperada a um pacote anteriormente enviado, ou pelo menos se o pacote diz ser de um host para o qual foi enviado recentemente um pacote. O firewall consegue identificar se uma mensagem de erro ICMP chegou em resposta a uma sessão corrente.

Firewalls dinâmicos normalmente têm conhecimento específico das aplicações ou protocolos. Por exemplo, um firewall pode fechar portas não privilegiadas e abrir uma porta para deixar passar uma sessão de FTP data.

1.3 CADEIAS E “CHAINS”

Atenção para os seguintes aspectos, essenciais para compreender os tópicos adiante sobre Iptables!!!



TOME NOTA!

O filtro de **input** e o filtro de **output** de uma interface tem regras separadas e independentes. As listas de regras que definem que tráfego é permitido e que tráfego não é permitido são chamadas de cadeias de regras (**chains**).



lista se acabar.

As regras de filtragem da firewall estão agrupadas em várias sub-listas com o propósito de separar tráfego de naturezas diferentes.

As sub-listas dividem o tráfego pela sua natureza: tráfego que se destina à firewall (**INPUT**), tráfego gerado pela firewall (**OUTPUT**) e tráfego que passa pela firewall, destinado a uma máquina interna, ou a um destino externo (**FORWARD**).

Cada cadeia de regras possui uma política por omissão e um conjunto de acções ou regras para seguir em resposta a tipos de mensagens específicas que entram na cadeia de regras.

Quando um pacote entra numa cadeia de regras é submetido a uma lista de regras até encontrar uma regra que se aplique ao pacote. Se o pacote não encontrar nenhuma regra que se aplique a ele, a política por omissão é aplicada ao pacote.

Existem duas posturas básicas, ou políticas por omissão que podem ser utilizadas:

- **Negar tudo** e explicitamente permitir certos pacotes.
- **Aceitar tudo** e explicitamente negar determinados pacotes.

Por norma, adota-se como política negar tudo por padrão. Deste modo é mais fácil configurar um firewall seguro.

Esta policy implica que é necessário explicitamente activar serviços, permitindo os protocolos utilizados nesses serviços. Implica ainda que é necessário conhecer os protocolos de comunicação de cada serviço que é ativado. Requer mais trabalho inicial na configuração mas é mais segura. Alguns produtos de firewalls comerciais só suportam esta política.

1.4 IPTABLES

O **Iptables** é um firewall que utiliza o mecanismo de Netfilter presente no Kernel Linux e normalmente é incluído em todas as distribuições de Linux.

O iptables pertence a um framework existente no Kernel Linux desde a versão 2.4.x e fornece mecanismos de **filtro-de-pacotes** (stateful), NAT e packet-mangling.

O **netfilter** é um mecanismo presente na implementação na pilha de rede (network stack)



O iptables possui uma estrutura de tabelas que definem conjuntos de regras. Cada regra dentro de uma tabela contém um conjunto de correspondências (matches) e um destino de pacote (target).

O iptables resulta de evoluções e reestruturações dos seus antecessores ipchains (presente nos Kernel versão 2.2.x) e ipfwadm (presente nos Kernel versão 2.0.x).

As **principais características** do Iptables incluem:

- Filtro dinâmico de pacotes (stateful);
- Todos os tipos de NAT;
- É possível configurar proxys transparentes;
- Manipulação de pacotes (packet mangling)
- Estrutura flexível e extensível.
- Existem muitas extensões disponíveis.

Atualmente, este firewall é o mais utilizada em ambiente Linux, e a diferença de performance para muitas outros firewalls é significativa.

Isto deve-se ao fato de a maior parte da firewall iptables correr em modo privilegiado do Kernel, evitando assim que os pacotes tenham que “subir” até às camadas superiores da pilha protocolar TCP-IP para serem filtrados.

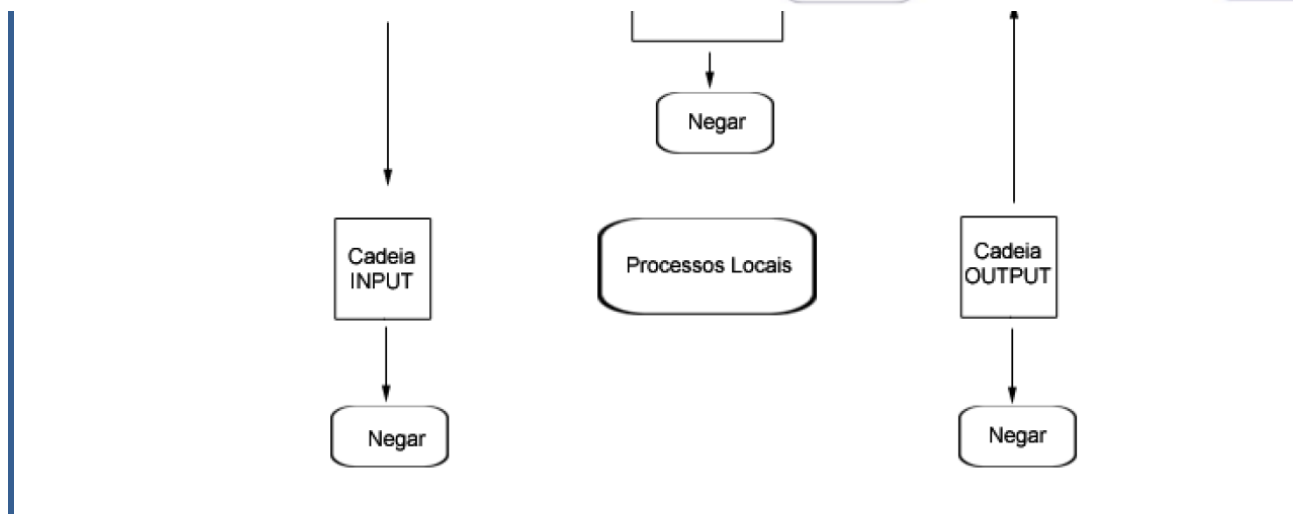
1.5 ADMINISTRAÇÃO IPTABLES

No iptables, são utilizadas três chains: **INPUT, OUTPUT e FORWARD**.

Os pacotes que chegam passam por um módulo de pré encaminhamento dos pacotes (routing), que determina se o pacote será entregue à cadeia de regras INPUT ou à cadeia de regras FORWARD.

O fluxo dos pacotes no iptables pode ser observado na figura abaixo.





Se um pacote que chegou ao firewall for aceito pela cadeia INPUT, o pacote é entregue localmente. Se o um pacote destinado remotamente é aceito pela cadeia FORWARD, o pacote é enviado para a interface apropriada.

Os pacotes de processos locais são passados á cadeia OUTPUT. Se o pacote for aceito, é enviado para a interface apropriada. Cada pacote é apenas filtrado uma vez (exceto os pacotes loopback que são filtrados duas vezes).

O iptables possui tabelas de regras para diferentes tipos de pacotes processados. Estas tabelas de regras são implementadas como módulos de funcionalidades separadas.

Os três módulos primários são a tabela de regras **filter**, a tabela para **Network Address Translation** e a tabela para manipulação de pacotes **mangle**.

1.6 TABELA FILTER

A **tabela filter** é a tabela utilizada por padrão. Para se especificar outras tabelas, é necessário utilizar um parâmetro na linha de comandos.

As características básicas da tabela filter incluem:

- Operações relacionadas com as três cadeias de regras base (INPUT, OUTPUT e FORWARD) e as cadeias definidas pelos utilizadores (user defined chains).
- Aceitar ou negar (descartar silenciosamente) um pacote (ACCEPT e DROP)
- Operações para encontrar matches no cabeçalho IP, do protocolo, dos endereços origem e destino, das interfaces de entrada e de saída e da manipulação de fragmentos.
- Operações para fazer match de informação presentes nos campos dos cabeçalhos TCP, UDP e ICMP.



1.7 TABELA NAT

Existem três funções gerais de utilização do NAT:

- **NAT tradicional**, unidireccional aplicado a pacotes enviados: Utilizado em redes que utilizam endereços privados. Subdivide-se em:
 - NAT básico: apenas tradução de endereço. Normalmente utilizado para mapear endereços privados com um conjunto de endereços públicos.
 - NAPT ou PAT (Network Address Port Translation): Normalmente utilizado para mapear endereços privados para um único endereço público (exemplo: Masquerading do Linux).
- **NAT bidireccional**: tradução bidireccional de endereços para pacotes que chegam ou que saiam da firewall. Uma possível utilização é conversão bidireccional entre endereços IPv4 e IPv6.
- **Nat duplo**: permite tradução bidireccional dos endereços origem e destino para traduzir os endereços origem e destino. Pode ser utilizado quando os endereços de rede origem e destino colidem, o ser resultado de por um lapso, alguém estar a utilizar um endereço IP pertencente a outra máquina. Também pode ser utilizado quando por conveniência os endereços de uma rede foram alterados e o administrador não quer alterar os endereços individualmente em cada máquina da rede.

A tabela nat possui módulos de extensão para tradução de endereços origem e destino e para **tradução de portas (PAT)**. Estes módulos suportam as seguintes formas de NAT:

- - **SNAT**: Source NAT
 - **DNAT**: Destination NAT
 - **MASQUERADE**: Uma forma especializada de NAT utilizado em ligações à Internet cujo IP é atribuído temporariamente de uma forma dinâmica. O estado de uma ligação é esquecido logo após a ligação se perder.
 - **REDIRECT**: Uma forma especializada de NAT que redirecciona um pacote para a máquina local, independentemente do endereço IP que este possui no campo destino do cabeçalho IP.

Tal como a tabela filter, a tabela NAT também possui 3 cadeias de regras base:

- A cadeia **PREROUTING** permite alterar o endereço destino dos pacotes antes de os passar a função de encaminhamento (routing) (Destination NAT). A alteração do destino pode ser para próprio firewall (proxy transparente, redirecionamento de portas) ou para outra máquina (masquerading, port forwarding em terminologia utilizada no Linux).
- A cadeia **OUTPUT** permite alterar o destino de um pacote fabricado no firewall, antes das decisões de encaminhamento (DNAT, REDIRECT). Isto é utilizado para transparentemente redirecionar um pacote de saída da firewall, para um proxy local, mas pode também ser utilizado para fazer port forwarding para uma máquina diferente.
- A cadeia **POSTROUTING** permite modificar o endereço origem de pacotes de saída do firewall (SNAT, Masquerade). As alterações são feitas depois da função de encaminhamento



A diferença entre **SNAT** regular e **MASQUERADE** é que em SNAT o estado da ligação é mantido até um período de timeout se esgotar.

Se uma ligação cair e for restabelecida rapidamente, os programas de rede podem continuar a funcionar sem perturbações visto o seu endereço IP não ter mudado, e qualquer tráfego TCP que possa ter sido interrompido, é retransmitido.

A figura abaixo lista as **operações básicas de match na tabela filter**.

Comando	Descrição
-i --in-interface [!] [<interface>]	Para pacotes nas cadeias de INPUT ou FORWARD, ou nas suas sub-cadeias definidas pelo utilizador. Especifica a interface a que a regra se aplica. Se não for especificada uma interface, a regra é aplicada a todas as interfaces.
-o --out-interface [!] [<interface>]	Para pacotes nas chains OUTPUT ou FORWARD, ou nas suas sub-cadeias definidas pelo utilizador. Especifica a interface a que a regra se aplica. Se não especificada uma interface, a regra é aplicada a todas as interfaces.
-p --protocol [!] [<protocolo>]	Especifica o protocolo IP a que a regra se aplica. As hipóteses permitidas são <i>tcp</i> , <i>udp</i> , <i>icmp</i> e <i>all</i> . O valor do parâmetro <i>protocolo</i> pode ser o nome ou o valor numérico, listado em <i>/etc/protocols</i>
-S --source --src [!] [<endereço>[/<máscara>]	Especifica o host ou rede do endereço origem do pacote, no cabeçalho IP.
-d --destination --dst [!] [<endereço>[/<máscara>]	Especifica o host ou rede do endereço destino do pacote, no cabeçalho IP.
-j --jump <target>	Especifica o <i>target</i> (destino de regras) para o pacote caso haja um <i>match</i> da regra. Os <i>targets</i> por defeito são o ACCEPT e o DROP, ou uma cadeia definida pelo utilizador.



A tabela mangle permite marcar ou associar um valor (mantido pelo Netfilter/iptables) a um pacote, e modificar o campo TOS (type of service) do pacote para casos particulares antes de ser feita a decisão de routing.

A tabela mangle tem duas cadeias de regras base:

- A cadeia **PREROUTING** permite modificar os pacotes à medida que cheguem à firewall, antes de passarem pela função de encaminhamento (routing).
- A cadeia **OUTPUT** permite modificar os pacotes gerados localmente.

Um firewall Linux pode ser configurado para respeitar a flag Type Of Service do cabeçalho TCP definida pela tabela mangle ou por uma máquina local.

Existe pouca informação sobre a marcação de pacotes na documentação do iptables. Essa marcação é utilizada na implementação do Linux QoS e que é utilizada como flag de comunicação entre módulos do iptables.

1.9 SCRIPTS ADMINISTRAÇÃO IPTABLES

O iptables é configurado por intermédio de comandos. Para cada regra de cada cadeia de regras a introduzir, é necessário um comando.

Tudo no iptables é configurado via comandos. Esses comandos são normalmente agrupados num ficheiro de script que é executado quando o servidor Linux inicia de modo a reconfigurar a firewall.

As regras utilizam muitos dados em comum como o endereço IP local, o nome da interface externa, etc. É prática comum definir esses dados como variáveis no início do script da firewall. As variáveis em shell scripts iniciam-se pelo prefixo \$.

Os comandos da tabela filter são fornecidos pelo módulo ip_tables. Esta funcionalidade é activada quando o módulo é carregado, o que acontece automaticamente na primeira invocação do comando iptables.

As principais operações com **chains** são:



	de utilizador.
-F --flush [<cadeia>]	Apaga o conteúdo de uma cadeia, ou o conteúdo de todas as cadeias, caso nenhuma tenha sido especificada.
-X --delete-chain [<cadeia>]	Apaga uma cadeia de utilizador, ou todas as cadeias caso nenhuma tenha sido especificada.
-P --policy <cadeia> <policy>	Define a política default para uma das cadeias base, INPUT, OUTPUT e FORWARD. As políticas possíveis são ACCEPT ou DROP.
-L --list [<cadeia>]	Listas as regras de uma cadeia, ou todas as cadeias caso nenhuma tenha sido especificada.
-Z --zero	Reinicia as contagens de bytes e pacotes associadas com cada cadeia.
-h <comando> -h	Lista os comandos iptables e as suas opções, ou se for especificado algum comando, lista a sintaxe e opções desse comando.

O comando **List** as seguintes opções adicionais:

Comando	Descrição
-L -n --numeric	Lista os endereços IP e portas numericamente, em vez de listar por nome.
-L -v --verbose	Lista informação adicional acerca de cada regra, como as contagens de bytes e pacotes, opções da regra, interfaces de rede relevantes, etc.
-L -x --exact	Lista os valores exactos das contagens, em vez de contagens arredondadas.
-L --line-numbers	Lista a posição da regra dentro da cadeia a que pertence.

A figura abaixo ilustra os comandos mais frequentes para criar e apagar regras dentro de uma cadeia.



-A --append <cadeia>	Adiciona uma regra ao fim de uma cadeia.
-I --insert <cadeia>	Adiciona uma regra ao início de uma cadeia.
-D --delete <cadeia> <número da regra>	Apaga um regra na posição indicada por parâmetro de uma cadeia.

A tabela filter permite fazer match de pacotes baseado em informação dos cabeçalhos TCP,UDP e ICMP, estado de ligação, listas de portas, acesso ao endereço MAC e ao campo TOS do cabeçalho IP.

Exemplos:

A regra seguinte **descarta pacotes UDP** que chegam a portas associadas ao serviço NFS e lockd:

```
# iptables -A INPUT -i eth0 -p udp -m multiport --destination-port 2049,4045 -j DROP
```

A regra seguinte **rejeita** (descarta silenciosamente) **ligações a máquinas remotas**, às portas associadas aos serviços NFS e squid:

```
# iptables -A OUTPUT -o eth0 -p tcp -m multiport --destination-port 2049,3128 --syn -j REJECT
```

A regra seguinte aceita ligações a máquinas remotas, às portas associadas aos serviços HTTP e HTTPS:

```
# iptables -A OUTPUT -o eth0 -p tcp -m multiport --destination-port 80,443 --syn -j ACCEPT
```



sobre os conceitos.

Mas além disso, devem ter sempre em mente é que a resolução de questões das bancas permite observarmos como as bancas abordam os tópicos e quais são os pontos preferidos do examinador.

Observar as tendências da banca é muito importante para facilitar e otimizar nossos estudos.



1.10 RESOLUÇÃO DE QUESTÕES

Pessoal, peço desculpas mas não foram identificadas questões da banca relativas ao nosso tópico de estudo.



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.