Introdução
As formas tradicionais de classificação de pacotes, baseadas geralmente no tipo de protocolo de transporte, na camada 4 do modelo de rede OSI (TCP ou UDP) e nas portas de origem e/ou destino, são suficientes na maioria dos casos. Porém ela pode ser ineficaz em alguns casos específicos, como por exemplo, no caso de programas de P2P (programas de compartilhamento de arquivos, como o Kazaa), que utilizam portas aleatórias ou em caso de serviços rodando em uma porta não padrão (por exemplo um servidor HTTP rodando na porta 1111). Em casos como este, o uso de um classificador de pacotes que faz a análise dos dados na camada de aplicação (camada 7 do modelo OSI) torna-se indispensável.

Funcionamento
O sistema atua como uma extensão do *iptables*, e é utilizado através da sintaxe "-m layer7 --l7proto XXXX", onde XXXX é o protocolo desejado (veremos alguns exemplos mais à frente). Vamos a um exemplo mais concreto:

 # iptables -I INPUT -m layer7 --l7proto fasttrack -j DROP

Este comando "derruba" (-j DROP) pacotes que utilizem o protocolo FastTrack (--l7proto fasttrack) e estejam entrando na máquina (-I INPUT). Vale lembrar que o protocolo FastTrack é utilizado pelo Kazaa e por outros aplicativos P2P. Perceba que não é feita nenhuma referência à portas ou endereços IP (embora elas possam ser utilizadas em conjunto - você poderia escrever essa regra de forma a afetar apenas um host especificamente).

Também é possível utilizar o L7-Filter em conjunto com a ferramenta tc, para fazer controle de banda.
Obs: o trecho acima foi retirado do site www.vivaolinux.com.br
Recompilando o kernel no fedoracore6 para aplicação do layer7

Antes de compilar o Kernel instale os pacotes abaixo, sem eles a compilação pode falhar, ja estou informando com o comando "yum install" para facilitar, são eles:

yum install hardlink
yum install kernel-devel
yum install kernel-doc
yum install glibc
yum install glibc-common
yum install glibc-headers
yum install glibc-devel
yum install cpp
yum install gcc
yum install libgcc
yum install ncurses-devel
yum install redhat-rpm-config
yum install rpm
yum install rpm-python

INSTITUTO DE DESENVOLVIMENTO SUSTENTÁVEL MAMIRAUA
COORDENADORIA DE INFORMÁTICA
AUTOR: Martinelli Souza
E-mail: martinelli@mamiraua.org.br

Esses pacotes são essenciais para funcionar o comando MAKE, eu testei todos com Fedora6 e a compilação foi um sucesso.

Primeiramente vamos baixar os pacotes necessários para a aplicação do layer7 e IPP2P

kernel-2.6.18-1.2798.fc6.src.rpm - http://download.fedora.redhat.com/pub/fedora/linux/core/6/source/SRPMS/kernel-2.6.18-1.2798.fc6.src.rpm
iptables-1.3.7 - http://www.netfilter.org/projects/iptables/files/iptables-1.3.7.tar.bz2
netfilter-layer7-v2.7.tar.gz - http://downloads.sourceforge.net/l7-filter/netfilter-layer7-v2.7.tar.gz?modtime=1165969076&big_mirror=0
l7-protocols-2006-12-12.tar.gz - http://downloads.sourceforge.net/l7-filter/l7-protocols-2006-12-12.tar.gz?modtime=1165919032&big_mirror=0

Antes de começarmos qualquer procedimento e tomando por base que você acabou de fazer uma instalação do fedoracore6 vamos remover o iptables com o comando abaixo:

yum remove iptables

Repare que estou utilizando um kernel do próprio fedora para que não haja maiores problemas., a versão abordada nesse tutorial esta como um pacote rpm então temos que retirar o pacotes linux-2.6.18., existem duas maneiras para a remoção se você estiver utilizando o ambiente gráfico basta apenas clicar com o contrário do mouse e pedir para extrair o pacote ., caso contrário execute o comando abaixo:

rpm -ivh kernel-2.6.18-1.2798.fc6.src.rpm

Feito isso ele descompactara o pacote em /usr/src/kernels

Descompactação dos pacotes
Vamos começar a compilação, primeiramente vamos descompactar os pacotes:
tar -zxvf iptables-1.3.7.tar.bz2 /usr/src **(se não obtiver êxito utilize a opção # tar - zxvf iptables-1.3.7.tar.bz2 -C /usr/src)**
tar -zxvf netfilter-layer7-v2.7.tar.gz /usr/src
tar -zxvf l7-protocols-2006-12-12.tar.gz /usr/src

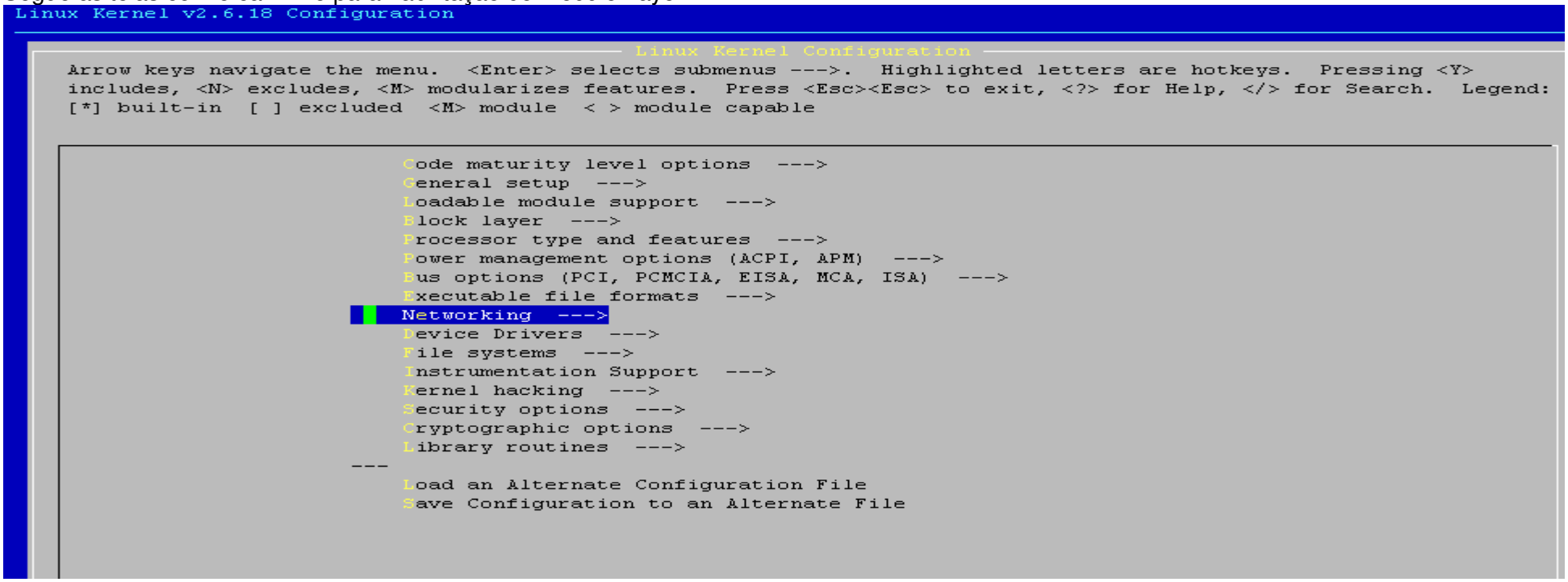
O pacote do kernel o linux-2.6.18 você pode mover /usr/src da forma que achar melhor ou mais fácil.

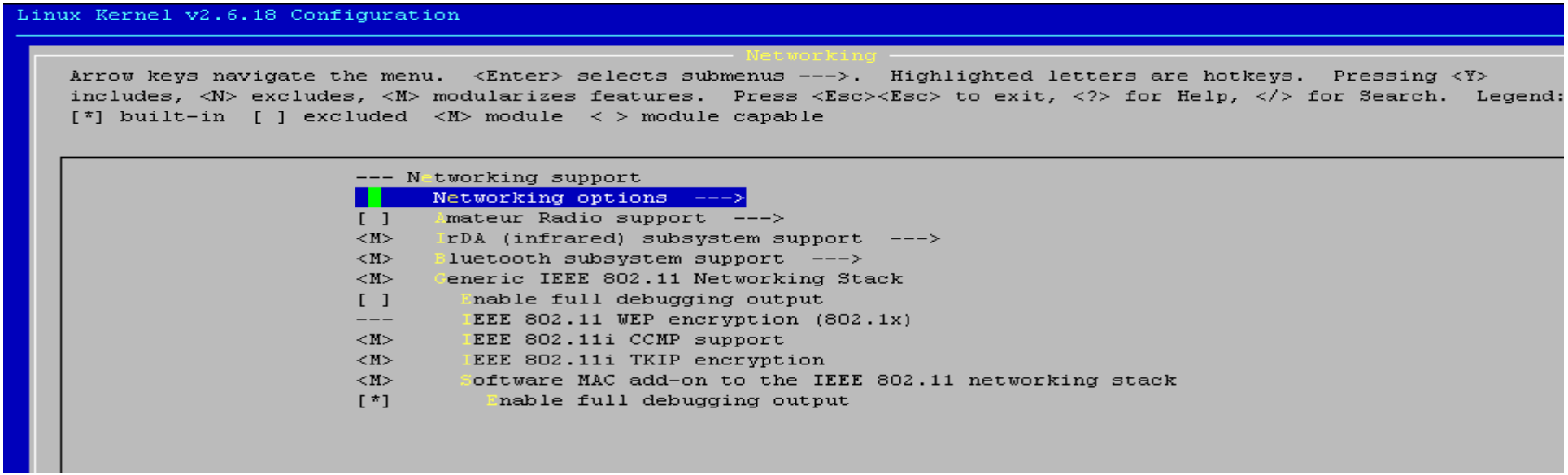
Temos agora que criar dois links simbólicos um para o pacote do iptables e outro para o pacote do kernel

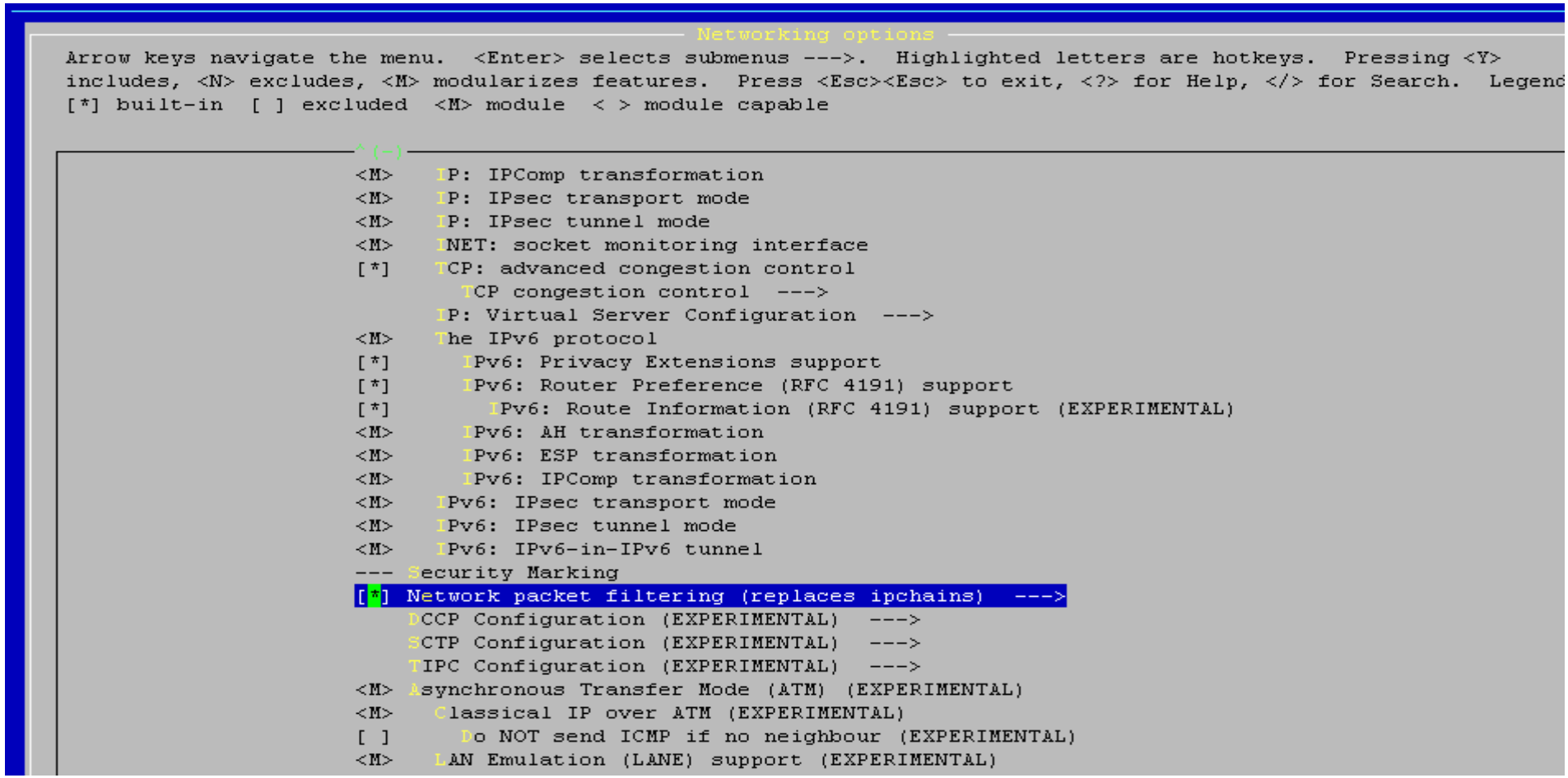
cd /usr/src
ln -s /usr/src/ iptables-1.3.7 /usr/src/iptables
ln -s /usr/src/linux-2.6.18 /usr/src/linux

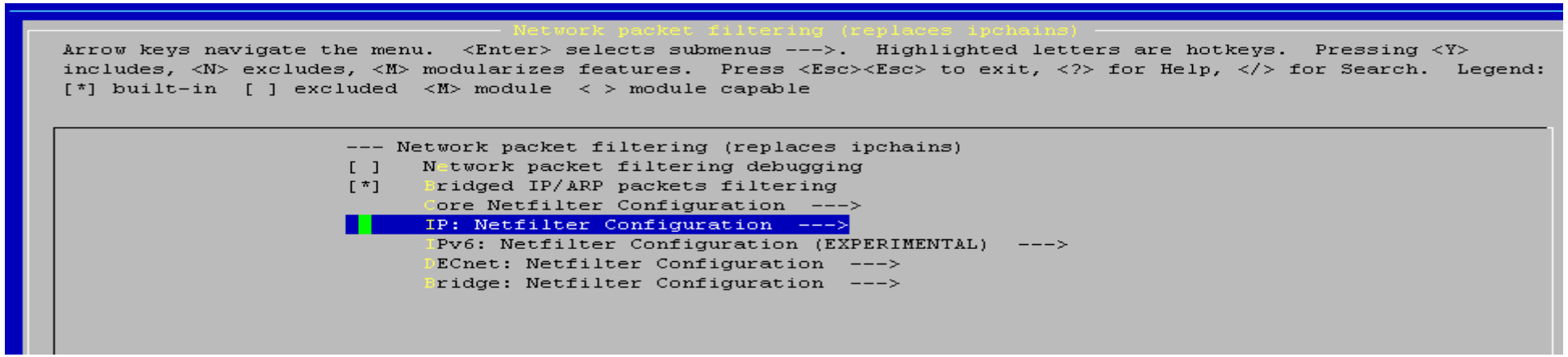
Ré-compilção para o novo kernel
cd /usr/src/linux
patch -p1 < /usr/src/ netfilter-layer7-v2.7/ kernel-2.6.18-layer7-2.7.patch
make menuconfig

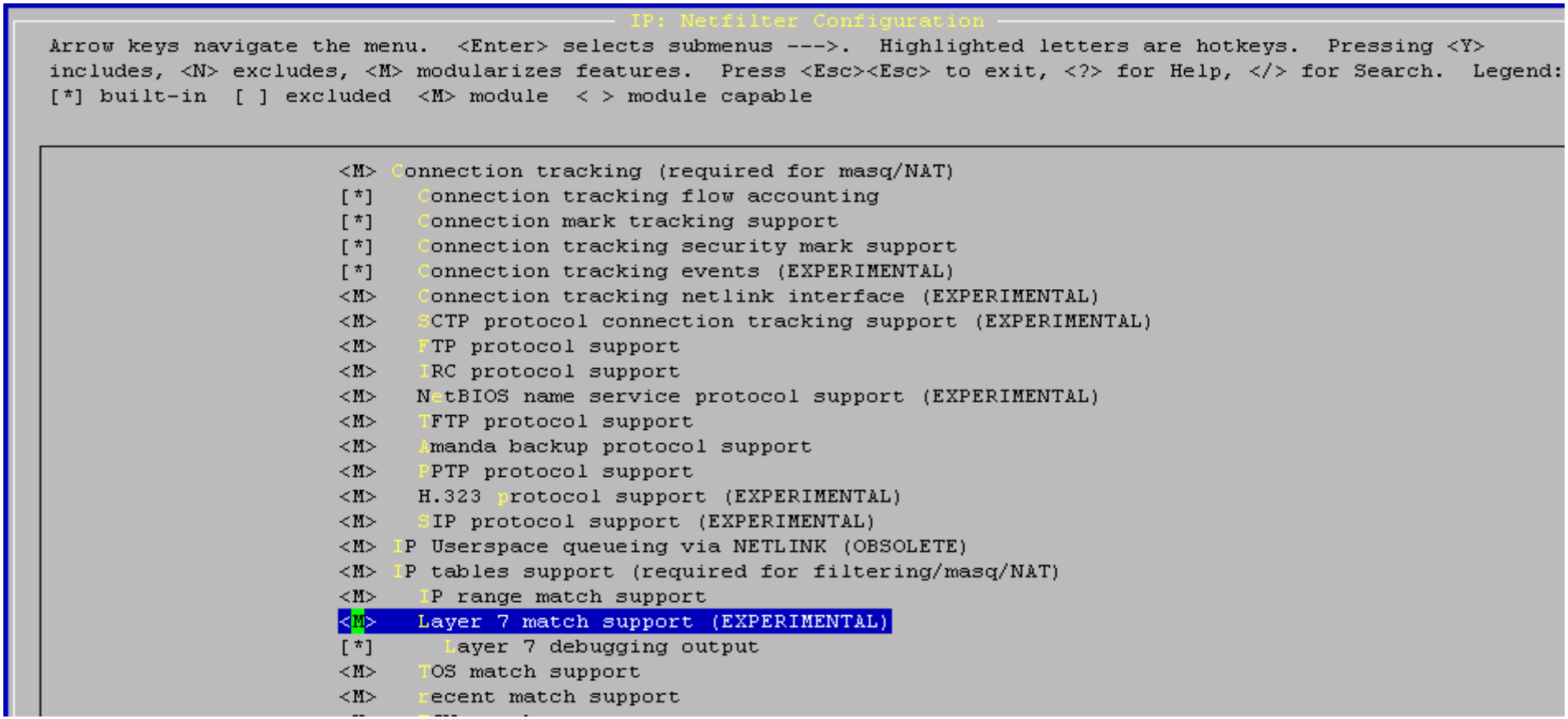
Segue as telas com o caminho para habilitação do módulo Layer7






Networking options
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend
[*] built-in [ ] excluded <M> module < > module capable
<M> IP: IPComp transformation
<M> IP: IPsec transport mode
<M> IP: IPsec tunnel mode
<M> INET: socket monitoring interface
[*] TCP: advanced congestion control
TCP congestion control --->
IP: Virtual Server Configuration --->
<M> The IPv6 protocol
[*] IPv6: Privacy Extensions support
[*] IPv6: Router Preference (RFC 4191) support
[*] IPv6: Route Information (RFC 4191) support (EXPERIMENTAL)
<M> IPv6: AH transformation
<M> IPv6: ESP transformation
<M> IPv6: IPComp transformation
<M> IPv6: IPsec transport mode
<M> IPv6: IPsec tunnel mode
<M> IPv6: IPv6-in-IPv6 tunnel
--- Security Marking
[*] Network packet filtering (replaces ipchains) --->
DCCP Configuration (EXPERIMENTAL) --->
SCTP Configuration (EXPERIMENTAL) --->
TIPC Configuration (EXPERIMENTAL) --->
<M> Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
<M> Classical IP over ATM (EXPERIMENTAL)
[ ] Do NOT send ICMP if no neighbour (EXPERIMENTAL)
<M> LAN Emulation (LANE) support (EXPERIMENTAL)


Network packet filtering (replaces ipchains)
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module < > module capable
--- Network packet filtering (replaces ipchains)
[ ] Network packet filtering debugging
[*] Bridged IP/ARP packets filtering
Core Netfilter Configuration --->
IP: Netfilter Configuration --->
IPv6: Netfilter Configuration (EXPERIMENTAL) --->
DECnet: Netfilter Configuration --->
Bridge: Netfilter Configuration -->



Marque a opção <M> Layer 7 match support (EXPERIMENTAL)
[*] Layer 7 debugging output

Depois salve as modificações e saia.
Agora siga os comandos abaixo ., isso vai demorar bastante tempo e se até aqui você usou os pacotes indicados e seguiu a risca o tutorial não haverá problemas.

make dep
make clean
make bzImage
make modules
make modules_install
make install

Depois disso reinicie a máquina.

Instalação do iptables-1.3.7

```
# cd /usr/src/iptables
# patch -p1 < /usr/src/ netfilter-layer7-v2.7/ iptables-layer7-2.7.patch
# chmod 755 extensions/ .layer7-test
# make KERNELDIR=/usr/src/linux
# make KERNELDIR=/usr/src/linux install
```

Cópia dos protocolos do layer7

Agora devemos copiar os arquivos dos protocolos do Layer7.
Primeiro crie o diretório l7-protocols dentro de /etc

```
# mkdir /etc/ l7-protocols
# cp /usr/src/l7-protocols-2006-12-12/protocols  /etc/l7-protocols
```

Lembre-se de verificar a versão desse arquivos (l7-protocols-2006-12-12) , pois ele está sempre sendo atualizado., quando mais novo melhor.
Para verificar que o layer7 foi perfeitamente instalado e aplicado ao iptables execute o comando abaixo., se retorna os que esta descrito é porque correu tudo ok.

```
# iptables -m layer7 --help

iptables v1.3.7

Usage: iptables -[AD] chain rule-specification [options]
       iptables -[RI] chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LFZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)

Commands:
Either long or short options are allowed.
  --append  -A chain            Append to chain
  --delete  -D chain            Delete matching rule from chain
  --delete  -D chain rulenum
                        Delete rule rulenum (1 = first) from chain
  --insert  -I chain [rulenum]
                        Insert in chain as rulenum (default 1=first)
  --replace -R chain rulenum
                        Replace rule rulenum (1 = first) in chain
  --list    -L [chain]          List the rules in a chain or all chains
```

```
  --flush   -F [chain]          Delete all rules in  chain or all chains
  --zero    -Z [chain]          Zero counters in chain or all chains
  --new     -N chain             Create a new user-defined chain
  --delete-chain
           -X [chain]          Delete a user-defined chain
  --policy  -P chain target
                               Change policy on chain to target
  --rename-chain
           -E old-chain new-chain
                               Change chain name, (moving any references)
Options:
  --proto       -p [!] proto    protocol: by number or name, eg. `tcp'
  --source      -s [!] address[/mask]
                               source specification
  --destination -d [!] address[/mask]
                               destination specification
  --in-interface -i [!] input name[+]
                               network interface name ([+] for wildcard)
  --jump        -j target
                               target for rule (may load target extension)
  --goto      -g chain
                              jump to chain with no return
  --match       -m match
                               extended match (may load extension)
  --numeric     -n              numeric output of addresses and ports
  --out-interface -o [!] output name[+]
                               network interface name ([+] for wildcard)
  --table       -t table        table to manipulate (default: `filter')
  --verbose     -v              verbose mode
  --line-numbers                print line numbers when listing
  --exact       -x              expand numbers (display exact values)
[!] --fragment  -f              match second or further fragments only
  --modprobe=<command>          try to insert modules using this command
  --set-counters PKTS BYTES     set the counter during insert/append
[!] --version   -V              print package version.

LAYER7 match v1.3.7 options:
--l7dir <directory>  : Look for patterns here instead of /etc/l7-protocols/
                       (--l7dir must be specified before --l7proto if used!)
--l7proto [!] <name> : Match the protocol defined in /etc/l7-protocols/name.pat
```

Para levantar o módulo do layer7 e só usar o comando abaixo
modprobe ipt_layer7

Agora crie as regras que acha necessário para fazer os bloqueios, no meu caso criei o script abaixo mas é somente uma pequena mostra do que o layer7 é capaz.

```
#!/bin/sh
#
echo "============================================================="
echo "                  Bloquear Softwares P2P
echo "============================================================
#
iptables -I FORWARD -m layer7 --l7proto edonkey -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto edonkey -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto fasttrack -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto fasttrack -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto directconnect -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto directconnect -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto bittorrent -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto bittorrent -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto napster -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto napster -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto soulseek -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto soulseek -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto gnutella -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto gnutella -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto msnmessenger -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto msnmessenger -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto imesh -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto imesh -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto ares -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto ares -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto counterstrike-source -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto counterstrike-source -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto doom3 -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto doom3 -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto dayofdefeat-source -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto dayofdefeat-source -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto halflife2-deathmatch -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto halflife2-deathmatch -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto quake-halflife -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto quake-halflife -s any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto quake1 -d any/0 -j DROP
iptables -I FORWARD -m layer7 --l7proto quake1 -s any/0 -j DROP

#
echo "============================================================="
echo "                  FIM DOS BLOQUEIOS
echo "============================================================
```