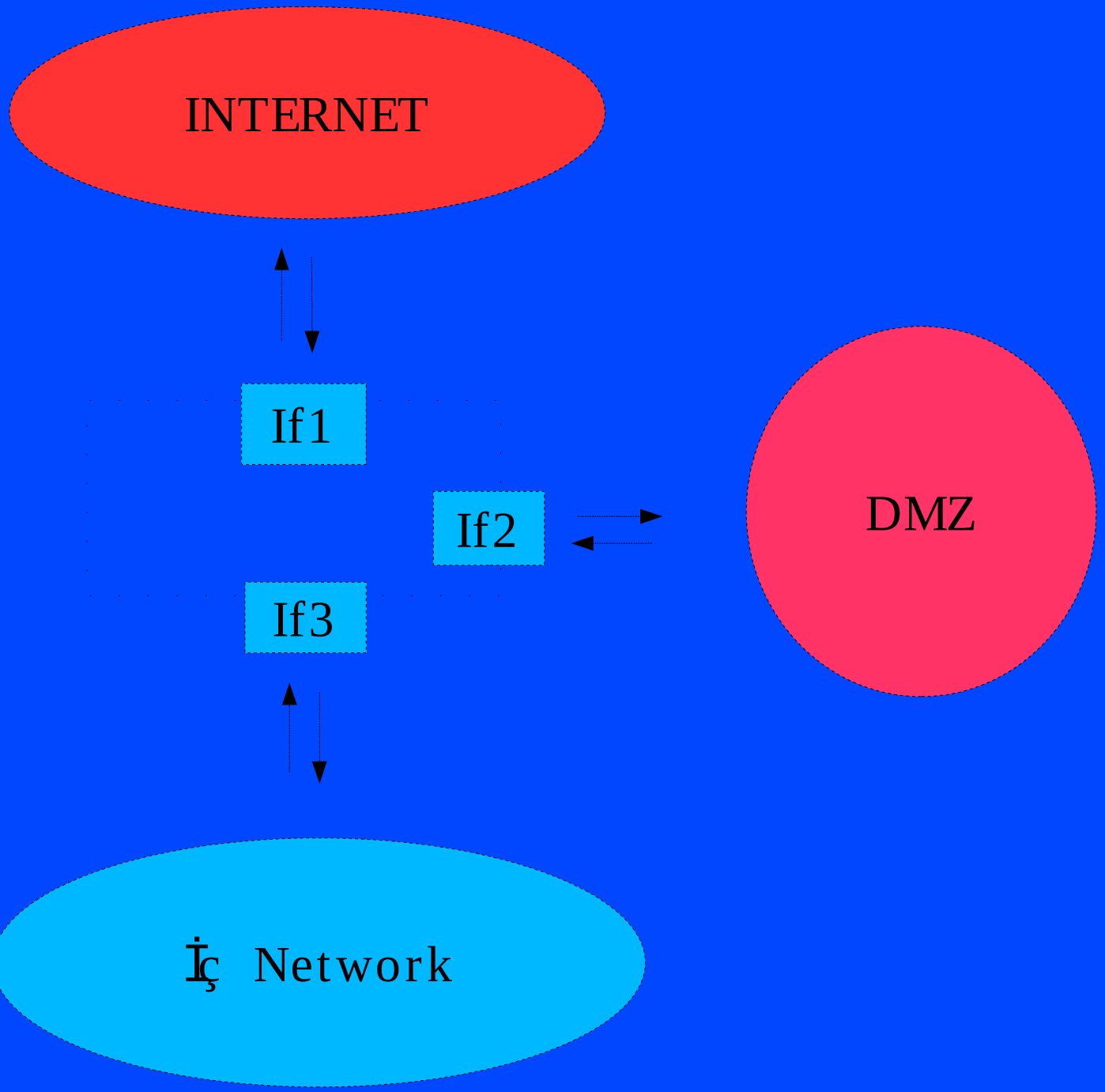
OpenBSD Packet Filter ile Görünmez Firewall Uygulaması

Devrim Sipahi
devrim.sipahi@deu.edu.tr

Yerleşimi



Görünmezlik için :

1. Ethernet kartlarına IP adresi verilmez.

2. Ethernet kartları köprü (bridge) modunda çalıştırılır.

3. Trafiğin geçmek zorunda olduğu bir kavşağa yerleştirilir.

Üstünlükleri

1. Yetkisiz kişiler tarafından görülmez.
2. Görülmediği için saldırılara hedef olmaz.
3. Network'te istenilen yere kolayca yerleştirilir.
4. Diğer cihazlarda (router) IP ayarlarıyla oynamaya gerek yoktur.
5. Devre dışı bırakmak için sadece kablo değiştirilir.
5. IP adres bloğunu bölmeye gerek yoktur.

OpenBSD Network ayarları

Ethernet kartları markasına göre isim alır:
xl0 : 3com ethernet
rl0 : realtek 8139
fxp0 : Intel ethernet
dc0 : Davicom ethernet
....

Aynı markadan birden fazla var ise 0,1,2 şeklinde son ek alırlar. xl0, xl1, xl2, rl0, rl1, rl2

OpenBSD Network ayarları

“/etc/hostname.if “ dosyası ethernet kartları ile ilgili yapılandırmaların yazıldığı dosyadır.
/etc/hostname.rl0
/etc/hostname.rl1
Bu dosyaların içinde sadece “up” yazması köprü (bridge) için yeterlidir.
“/etc/bridgename.bridge0” dosyasında köprü ile ilgili ayarlar bulunur.
add rl0
add rl1
up

Packet filter yapısı

PF bir kural dosyası (/etc/pf.conf) ve bir programdan (pfctl) oluşur. PF'i etkinletirmek için
pfctl -e
Devre dışı bırakmak için de
pfctl -d
komutu yeterlidir.
Sistemin açılışında etkin olması için /etc/rc.conf dosyasında
pf=YES
satırının bulunması yeterlidir.

Pfctl komutu uygulama seçenekleri

pf.conf dosyasındaki değişiklikleri yüklemek için
pfctl -f /etc/pf.conf

Yazım hatalarını kontrol için
pfctl -nf /etc/pf.conf

Sadece NAT kurallarını yüklemek için
pfctl -Nf /etc/pf.conf

Sadece süzgeç kurallarını yüklemek için
pfctl -Rf /etc/pf.conf

Pfctl komutu gösterme seçenekleri

NAT kurallarını görmek için
pfctl -sn

Süzgeç kurallarını görmek için
pfctl -sr

Durum tablosunu görmek için
pfctl -ss

Süzgeç durumunu ve sayacı görmek için
pfctl -si

Herşeyi görmek için de
pfctl -sa
komutu yeterlidir.

Paket Filtreleme

Filtre kurallarının genel yazım şekli:

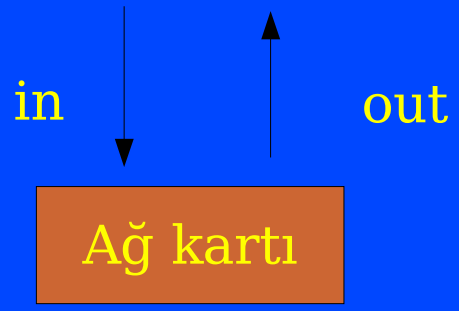
Karar doğrultu [log] [quick] on ağ_kartı [af] [proto protokol] \
from kaynak_adresi [port kaynak_portu] to hedef_adresi \
 [port hedef_portu] flags [tcp_flags] [state]

[ ] : Seçimlik
Siyah yazılanlar olduğu gibi kalacak,
Renklilerin yerine karşılıkları yazılacak.

Filtre açıklamaları

Karar : İlgili paketin geçmesine izin verilir (pass) ya da engellenir (block). block kararı “drop” ve “return” seçeneğiyle birlikte kullanılabilir.

Doğrultu : Paketin ağ kartına gelişi (in) veya gidişi (out)



Filtre açıklamaları

Ağ_kartı : Paketin geçtiği ağ kartı. rl0, xl0 vs..

af : Adres ailesi. Ipv4 için **inet**, Ipv6 için **inet6** yazılır.

protokol : 4. katman protokolleri: tcp, udp, icmp, icmp6 vs.

kaynak_adresi, hedef_adresi : Bu kısım bir Ipv4 veya Ipv6 adresi, ağ bloğu, makine adı, any, “from any to any” yerine all olabilir.

kaynak_portu, hedef_portu : 1 ile 65535 arası bir sayı, “/etc/services” dosyasındaki bir servis adı, bir aralık olabilir.

Filtre açıklamaları

tcp_flags : TCP başlığındaki bayraklara bakar. Örneğin “flags S/SA” Eğer SYN bayrağı aktif ise S ve A (SYN ve ACK) bayraklarına bakar.

state: Kurala uyan durum bilgisini tutar.

keep state : TCP, UDP ve ICMP ile birlikte çalışır.

modulate state : Sadece TCP ile çalışır. Kurala uymak için ISN sayısı üretir.

synproxy state : Sunucuları TCP SYN saldırılarına karşı korur. Köprü modunda çalışmaz.

Kurallardaki sıralama

Bir paketle ilgili SON uyan kural uygulanır.
Bu durumun istisnası “quick” sözcüğüdür.

Yanlış:
block in on rl0 proto tcp from any to any port ssh
pass in all

Doğrusu:

block in **quick** on rl0 proto tcp from any to any port ssh
pass in all

Yazım Kolaylıkları - Listeler

Liste kullanımı: Kural cümlesinde bir veri yazılması gereken yere küme parantezi ile istenilen sayıda veri yazılabilir.

```
block out on fxp0 proto { tcp udp } from { 192.168.0.1, \
   10.5.32.6 } to any port { ssh telnet }
```

Bu satır aşağıdaki 8 satırın işini yapar.

```
block out on fxp0 proto  tcp from 192.168.0.1 to any port ssh
block out on fxp0 proto  udp from 192.168.0.1 to any port ssh
block out on fxp0 proto  tcp from 10.5.32.6 to any port ssh
block out on fxp0 proto  udp from 10.5.32.6 to any port ssh
block out on fxp0 proto  tcp from 192.168.0.1 to any port telnet
block out on fxp0 proto  udp from 192.168.0.1 to any port telnet
block out on fxp0 proto  tcp from 10.5.32.6 to any port telnet
block out on fxp0 proto  udp from 10.5.32.6 to any port telnet
```

Yazım Kolaylıkları - Makrolar

Makrolar: Kullanıcı tarafından tanımlanan değişkenlerdir.

```
ext_if=”xl0”
block in on $ext_if from any to any
```

Ethernet kartınız değiştiğinde bütün satırlara bakmak yerine makroyu değiştirmek yeterlidir.

```
win_ports="{ 135, 137, 138, 139, 445, 446, 447, 448 }"
block in on $ext_if   proto { tcp udp } from any to any \
 port $win_ports
```

Yazım Kolaylıkları - Tablolar

Tablolar: IPv4 ve/veya IPv6 adresleri için kullanılır. İki farklı seçenekle kullanılabilir.
const: Tablo oluşturulduktan sonra değiştirilemez.
persist: Tabloyu bellekte tutar.
Örnek:

```
table <localnet> {192.168.0.0/24}
table <rfc1918> const {192.168.0.0/16, 172.16.0.0/12, \
 10.0.0.0/8}
block in on $ext_if from <rfc1918> to any
pass in on $int_if from <localnet> to any
```

Yazım Kolaylıkları - Tablolar

Tablolar: IP adreslerini belirtmek için bir text dosyası da kullanılabilir.

```
table <spammers> persist file "/etc/spammers"
table <SMTP> persist file "/etc/smtp"
block on $int_if proto tcp from ! <SMTP> to any port 25 flags S/SA
```

/etc/smtp dosyasının içinde IP adresleri altalta yazılır.

```
1.2.3.5
1.2.3.7
1.2.3.24
```

Adres grupları oluşturulabilir.

```
table <ornek> { 10.1.0.0/16, !10.1.1.0/24, 10.1.1.100 }
block in on xl0 all
pass in on xl0 from <ornek> to any
```

pfctl ile tablo işlemleri

Tabloya bir adres eklemek için
pfctl -t spammers -Tadd 20.200.1.3
Ayrıca bu komut bu isimde tablo yoksa oluşturur.

Tablonun içeriğini görmek için
pfctl -t spammers -Tshow

Tablodan bir IP grubu çıkarmak için
pfctl -t spammers -Tdelete 213.50.0.0/16

Kısaltmalar

Başlangıçta herşeyi yasaklamak için iki satır yazılır.

```
block in  all
block out all
```

Bunun yerine

```
block all
```

yazılabilir.

“from any to any“ de kısaltılabilir.

```
pass  in quick log on rl0 proto tcp from any to any port 22
pass  in quick log on rl0 proto tcp to port 22
```

Kısaltmalar

Return seçeneği kısaltmaları:
Engellenen TCP paketleri için “TCP RST”, diğer paketler için “ICMP Unreachable” cevabı gönderilir.

```
block in all
block return-rst in proto tcp all
block return-icmp in proto udp all
block out all
block return-rst out proto tcp all
block return-icmp out proto udp all
```

Bu satırların yerine

```
block return
```

yazmak yeterlidir.

Paketlerin normalleştirilmesi (Scrubbing)

Bununla bölünmüş paketler birleştirilir, geçersiz bayraklı TCP paketlerini atar ve bazı işletim sistemlerini bu tür ataklara karşı korur.
scrub in all

Seçenekler:
no-df : IP paket başlığından “don't fragment” bitini temizler.
random-id: Giden paketlerin IP tanıtım alanını rasgele değerlerle değiştirir.
min-ttl num: En küçük TTL değerini belirler.
max-mss num: Paket başlığındaki enbüyük Maximum Segment Size değerini belirler.
fragment reassemble: Gelen bölünmüş paketleri filtre kurallarını uygulamadan önce birleştirir.
Örnek:
scrub in on fxp0 all fragment reassemble min-ttl 15 max-mss 1400

Kayıt tutma ve anlık izleme

Firewall kayıtları pflogd(8) programı ile pflog0 arayüzü dinlenerek /var/log/pflog isimli dosyada tutulur. Bu dosya tcpdump binary formatındadır.
Kayıt tutulacak kurallarda **log** veya **log-all** sözcüğü bulunmalıdır. **log** sözcüğü sadece içinde keep state, modulate state, veya synproxy state sözcüğü bulunan satırları kaydeder.
Diğerlerini de kaydetmesi için **log-all** kullanılır.

Kayıtları görmek için
tcpdump -n -e -ttt -r /var/log/pflog

Gelen giden paketleri anlık izlemek için ise
tcpdump -n -e -ttt -i pflog0
komutu kullanılır.

Örnek kurallar

```
ext_if="rl0"
int_if="rl1"
table <SY> { 193.140.151.18, 193.140.151.19, 193.140.151.20 }
table <SMTP> persist file "/etc/fw/smtp"
WIN_PORTS="{ 135, 137, 138, 139, 445, 446, 447, 448, 464 }"
# Sadece sistem yoneticileri fw'a ssh ile baglanabilir.
block return in log-all on { $ext_if $int_if } proto tcp from ! <SY>  to
{ $ext_if $int_if }  port ssh flags S/SA
#Windows pc'lerin koruması
block return in  log-all on $ext_if   proto { tcp udp } from any to any  port
$WIN_PORTS

# Yetkili SMTP sunucular dışındakilerin Internet mail göndermemesi
block return in on $int_if proto tcp from ! <SMTP> to any port 25 flags
S/SA
```

Bandgenişliği Yönetimi

Bandgenişliği yönetimi bir kuyruk yönetimidir.

Sadece giden paketlere uygulanır.

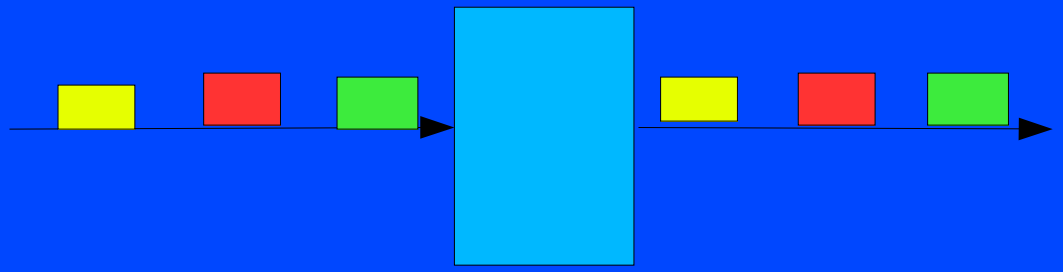
Normal kuyrukta FIFO, yani ilk gelen ilk gider kuralı uygulanır.

İki çeşit kuyruklama yapılabilir:

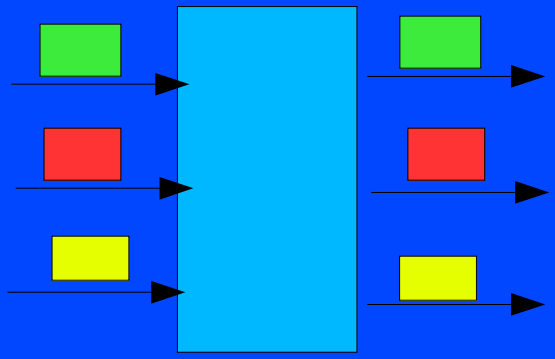
1. Sınıflandırılmış kuyruk (Class Based Queueing)

2. Torpilli kuyruk (Priority Queueing)

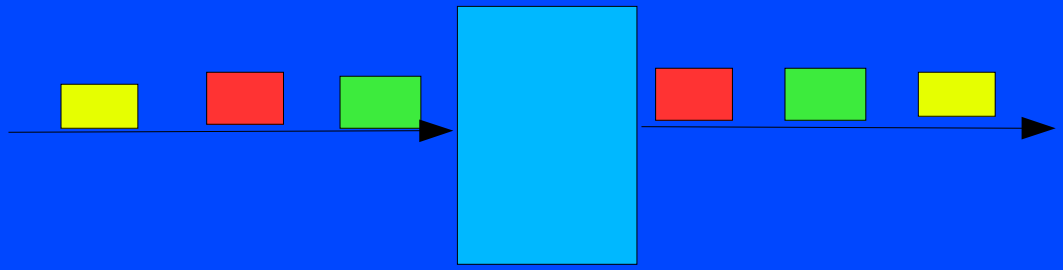
Bandgenişliği Yönetimi



Normal kuyruk



Sınıflandırılmış kuyruk



Torpilli kuyruk

Sınıflandırılmış kuyruk (CBQ)

KUYRUK YAPISININ PLANLANMASI

Root Queue (2Mbps)
- UserA (1Mbps)
  - ssh (50Kbps)
  - bulk (950Kbps)
- UserB (1Mbps)
  - audio (250Kbps)
  - bulk (750Kbps)
    - http (100Kbps)
    - other (650Kbps)

Sınıflandırılmış kuyruk (CBQ)
EK öZELLİKLER

ÖDÜNÇ ALMA

Root Queue (2Mbps)

UserA (1Mbps)

ssh (100Kbps)
ftp (900Kbps, borrow)

UserB (1Mbps)

ÖNCELİK VERME

Root Queue (2Mbps)

UserA (1Mbps, priority 1)

ssh (100Kbps, priority 5)
ftp (900Kbps, priority 3)

UserB (1Mbps, priority 1)

Torpilli kuyruk (PRIQ)

Önceliği yüksek olan paketlerin işi önce görülür.
Önceliği düşük paketlerin atılma riski vardır.

Root Queue (2Mbps)

- Queue A (priority 1)
- Queue B (priority 2)
- Queue C (priority 3)

Kuyruk yapılandırması

Yazım şekli:

```
altq on interface scheduler bandwidth bw qlimit qlim \
   tbrsize size queue { queue_list }
```

interface : rl0
scheduler: cbq veya priq (sadece biri kullanılabilir)
bandwidth: 100Mb veya %50
qlim: kuyruktaki en büyük paket sayısı (belirtilmezse 50)
size: Ethernet hızını ayarlayan miktar
(belirtilmemişse ethernet hızı)
queue_list: alt kuyruklar

Kuyruk yapılandırması

Yazım şekli:

queue name [on interface] bandwidth bw [priority pri] \
[qlimit qlim] scheduler (sched_options) { queue_list }

pri: kuyruk öncelik numarası
sched_options: default, red, rio, borrow, ecn olabilir.

Kuyruk yapılandırması

Yazım örneği:

```
altq on fxp0 cbq bandwidth 2Mb queue { std, ssh, ftp }

queue std bandwidth 50% cbq(default)
queue ssh { ssh_login, ssh_bulk }
  queue ssh_login priority 4 cbq(ecn)
  queue ssh_bulk cbq(ecn)
queue ftp bandwidth 500Kb priority 3 cbq(borrow red)

pass out on fxp0 from any to any port 22 queue ssh
```

Kuyruk yapılandırması

Yazım örneği 2:

```
altq on fxp0 cbq bandwidth 2Mb queue { std, ftp }
queue std cbq(default)
queue ftp bandwidth 1.5Mb

pass in on dc0 from any to any port 21 queue ftp
```

Yük dengeleme (sunucular için)

Bir sunucuya gelen istekleri dengeli bir şekilde
gerekli sayıda sunucuya dağıtmak için kullanılır.
Örnek (ders kayıt zamanında denenmiş) /etc/pf.conf dosyası

```
ext_if="rl0"
int_if="rl1"
web_ip="193.140.151.17"
web_servers = "{ 10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4, 10.1.1.5,
10.1.1.6}"

# web sunucunun varsa dış bağlantıları için gerekli satır
nat on $ext_if from $int_if:network to any -> ($ext_if)

# web suncuya gelen yükü dağıtmak için gerekli satır
rdr pass on $ext_if proto tcp from any to $web_ip port 80 ->
$web_servers round-robin sticky-address
```

Packet Filter'in diğer yetenekleri

NAT
Yönlendirme
Alt kurallar koyma (anchor)
Adres havuzu kullanma
Yük dengeleme (Internet hatları için)
Paket markalama
Authpf

Kaynaklar

1. http://www.openbsd.org/faq/pf/
2. http://www.inebriated.demon.nl/pf-howto/html/
3. http://www.benzedrine.cx/pf.html
4. http://www.obfuscation.org/ipf/
5. OpenBSD man sayfaları
pfctl(8), pf.conf(5), pf(4)
6. OpenBSD Türkçe kurulum dökümanı
http://www.enderunix.org/docs/openbsd_install.html
7. IPFILTER Türkçe dökümanı
http://www.enderunix.org/docs/ipfilter.html
8. Bu dökümana aşağıdaki adresten erişebilirsiniz.
http://www.tasarruf.net/seminer/openbsd_pf.sxi
http://web.deu.edu.tr/izmirunix/seminer/openbsd_pf.sxi