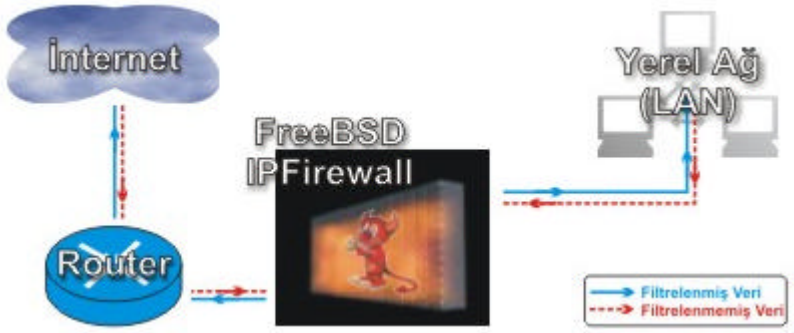
FreeBSD sistemlerde

IPFIREWALL Kurulumu ve Konfigürasyonu

Güncellenme tarihi: 8 Aralik 2002

IPFIREWALL(IPFW), isminden de anlasildigi gibi ip based bir paket filtreleme sistemidir. IPFIREWALL'u diger güvenlik duvarlarindan ayiran en büyük özelligi ek bir yazilima ihtiyaç duymadan traffic shaping yapabilmesidir. IPFIREWALL, tamamen kernel'a dayanir. Hatta tcpdump ile geçen paketleri sniff ettikten sonra en altta *???? packets received by filter, ???? packets dropped by kernel* ifadesi geçer. IPFWye bir kural eklediginiz zaman IPFWyi yeniden baslatmak gerekmez çünkü yeniden baslatamazsiniz zaten :) Kural çalistirildigi anda uygulanir.

* GÜVENLIK DUVARININ AG ÜZERINDEKI YERI:



Firewall ag üzerinde Router ile Yerel Ag arasinda ag geçidi olarak yer alir. Burada dikkat edilmesi gereken nokta clientlarda ag geçidi olarak router'i degil, firewallun tanimlanmasi. Firewall'un ag geçidi ise router olarak tanimlanmalidir.

---

* GEREKLI DONANIM:

Öncelikle, Yerel aginizin büyüklügüne göre bir bilgisayar seçmelisiniz. Sekilde görüldügü gibi Firewalla iki ayri baglanti baglaniyor. Bunun için 2 adet ethernet kartina ihtiyacimiz olacak. Yalniz ethernet kartinin markasinin seçiminde çok dikkatli olmamiz gerekiyor. Çünkü bütün internet trafigi bu ethernetlerden geçecegi için trafigi mümkün oldugunca hizli iletmeliyiz. En çok tercih edilen marka *Intel*.

Ethernetinizin markasi Intel ise, FreeBSD ethernet kartlarinizi **fxp0** ve **fxp1** olarak taniyacaktir. Router'a baglanan ethernet fxp0, Yerel Ag'a baglanan ethernet ise fxp1 olsun. fxp0'a size verilen registered IPlerden birini atayin. fxp1'e ise yerel aginizin büyüklügüne göre bir unregistered IP blogu seçip bir IP atayin. Örnek olarak fxp0'a 212.152.10.5 nolu ipyi, fxp1'e ise 192.168.0.1 nolu ipyi verdigimizi kabul ediyorum.

---

* BASLANGIÇ:

Ag ayarlarimiz tamamlandiktan sonra sistemi *cvsup* ile stable hale getirin.

Ilk yapmamiz gereken sey ethernet kartlari arasinda paket geçisine izin vermek. Bunun için */etc/sysctl.conf* dosyasina **net.inet.ip.forwarding=1** satirini ekleyin.

Açilis scriptlerinde düzenlemeler yapmamiz gerekiyor. Burada kullanmadigimiz tüm daemonlari kapatmaliyiz. Bunun için */etc/rc.conf* dosyasina asagidaki satirlari ekleyin veya var olanlari degistirin:

```
firewall_enable="YES"
firewall_type="open"
gateway_enable="YES"
tcp_drop_synfin="YES"
portmap_enable="NO"
inetd_enable="NO"
usbd_enable="NO"
```

---

* KERNEL:

IPFW'nin kernel tabanli oldugundan bahsetmistim. Gerekli tüm binary dosyalar FreeBSD kurulurken yükleniyor. Ancak kernel ayarlari yapilmadan kullanmayi denediginizde *"ipfw: getsockopt(IP_FW_GET): Protocol not available"* diye bir hata alirsiniz. Yani gerekli fonksiyonlar mevcut degil. bu durumda kernel'da bir takim degisiklikler yapacagiz.

Eger kernel'in ismini degistirmediyseniz GENERIC olarak sisteminizde yer almali. Asagidaki satirlari */usr/src/sys/i386/conf/GENERIC* isimli dosyaya ekleyin.

```
options ICMP_BANDLIM            #ICMP hata cevaplarina band genisligi sinirlamasi
                                getirir. D.O.S. attacklardan korunmaya yardimci olur.
options IPFIREWALL              #Firewall'i aktif eder
options IPFIREWALL_VERBOSE      #Log tutma (syslogd)
options
IPFIREWALL_VERBOSE_LIMIT=200    #Kural basina düsen log limiti
options IPFIREWALL_FORWARD      #TransparentProxy destegi
options IPDIVERT                #Port Yönlendirme
options DUMMYNET                #Traffic Shaper. Band genisligi ve kuyruk(queue)
                                ayarlari için gereklidir.
options IPSTEALTH               #Bu firewalldan geçen paketlerin TTL'lerinin
                                degistirmeden geçmesini saglar. Firewall'u traceroute
                                gibi araçlardan gizlemek için gereklidir.
options TCP_DROP_SYNFIN         #SFN+FINi düsür
```

Asagida kerneli nasil derleyip yükleyecegimizi basit olarak anlattim:

```
console# /usr/sbin/config GENERIC
console# cd ../../compile/GENERIC
console# make depend
console# make
console# make install
```

Kernel'i install ettikten sonra bilgisayarinizi yeniden baslatin.

Uyari: *Eger KernelSecurityLevel 2 veya daha yukarisi ise make install islemini single-user mod'a geçip öyle yapmaniz gerekiyor.*

---

* SÖZDIZIMI ve KULLANIM SEKLI:

Sözdizimi çesitleri:

```
ipfw [-f | -q] flush
```

```
ipfw [-q] {resetlog | delete} [kuralno ...]
ipfw [-q] add [kuralno] <rule-body>
ipfw pipe <kuralno> config <opsiyonlar>
ipfw pipe {delete | list | show} [kuralno ...]
```

Parametreler:

- **-a** Listeleme esnasinda sayaçlari gösterir. *show* parametresi default olarak -a parametresini barindirir.
- **-f** *flush* parametresi için onay istemez. Ancak ifade herhangi bir tty'den çalistirilmamissa -f default olarak uygulanir (Ör: flush istegi crontab'tan çagrilmissa herhangi bir tty'ye bagli olmadigi için -f default olarak uygulanir.)
- **-t** Kurallarin son eslesme zamanini gösterir.(timestamp)
- **-q** Yapilan islemlerin çiktisinin ekrana verilmesini engeller (sessiz)
- **-N** Kurallari listelerken IP adreslerini ve servisleri çözümleyip(resolve) göstermeye çalisir.

Kural düzeni:

```
ipfw add [kuralno] [prob match_probability] <action> [log [logamount number]]
<proto> from <src> to <dst> [interface-spec] [options]
```

<actions>:

`allow`	kurala uyan paketlerin geçmesine izin verir
`count`	kurala uyan paketlerin sayaçlarini günceller
`deny`	kurala uyan paketleri siler.
`divert` *`port`*	kurala uyan paketi verilen port'a yönlendirir. (bu action için kernel'a IPDIVERT seçeneginin yüklenmesi gerekir)
`fwd` *`ip [,port]`*	kurala uyan paketleri verilen ip ve port numarasina yönlendirir. IP adresi local bir IP ise port dikkate alinir. Ancak IP local degilse port numarasi dikkate alinmaz ve *route* ile *routing tablosu*na göre yönlendirilir.
`pipe` *`pipe_no`*	Paketi *dummynet*e yönlendirir. Dummynet bir trafik düzenleme kütüphanesidir. (Bandgenisligi, kuyruk ayarlari, paket bekletme vb.)
`reset`	kurala uyan paketleri reddeder. TCP reset (RST) paketi göndermeye çalisir.
`skipto` *`kuralno`*	kurala uyan paketleri verilen kural numarasina gönderir. Paket kurallardan geçmeye devam eder.
`tee` *`port`*	paketlerin birer kopyasini verilen port numarasina gönderir.
`unreache` *`kod`*	gelen paketleri reddeder ve cevaben ICMP unreach paketi gönderir. ICMP unreach paketi ise belirlenen kod numarasi ile belirlenir. kod numarasi 0 ile 255 arasindadir. En çok bilinen kodlar: net, host, protocol, port, needfrag, srcfail, net-unknown, host-unknown ...

<proto>: Kuralin protokolü (ip, tcp, udp veya all)

<src> to <dst>:

{any | me | [not] <ip adresi/maske> [portlar]}
any: tüm paketleri kapsar

me: Sistemin ethernet arayüznün ip numarasina uyan paketleri kapsar

<ip adresi/maske> asagidaki sekillerde kullanilabilir:

ipno: 1.2.3.4 seklinde olan ip adresleridir. Porta ve prokole bakilmaksizin bu ipyi içeren tüm paketleri kapsar
ipno/bit: 1.2.3.4/24 biçiminde olan ip adresleridir. Bu ifade 1.2.3.0 dan 1.2.3.255'e kadar olan tüm ip adreslerini kapsar
ipno:mask: 1.2.3.4:255.255.240.0 biçiminde olan ip adresleridir. Bu ifade 1.2.3.0 dan 1.2.15.255'e kadar olan tüm ip adreslerini kapsar

[portlar]: {port | port-port} [,port[,...]] biçiminde kullanilir. *port-port* ifadesi birinci port ile ikinci port arasindaki tüm portlari kapsar

`[interface-spec]:`

in: Sadece gelen paketleri kapsar
out: Sadece giden paketleri kapsar
via <*ifX*>: Sadece <*ifX*> arayüzünden geçen paketleri kapsar **via any**: Herhangi bir arayüzden geçen tüm paketleri kapsar

`[options]:`

established: RST veya ACK bitlerini içeren TCP paketlerini kapsar
icmptypes <*types*>: ICMP tipi, *types* parametresinde yer alan ICMP paketlerini kapsar. Deskteklenen ICMP tipleri asagida listelenmistir.

0	echo reply
3	destination unreachable
4	source quench
5	redirect
8	echo request
9	router advisement
10	router solicitation
11	time-to-live exceeded
12	IP header bad
13	timestamp request
14	timestamp reply
15	information request
16	information reply
17	address mask request
18	address mask reply

iplen <*uzunluk*>: Toplam (header+data) uzunlugu *uzunluk* kadar olan IP paketlerini kapsar.
ipoptions <*spec*>: IP Header'inda *spec* parametresinde yer alan özellikleri tasiyan paketleri kapsar. <*spec*> parametresinin alabilecegi degerler asagida listelenmistir.

ssrr	strict source route
lsrr	loose source route
rr	record packet route
ts	time stamp

ipttl <***ttl***>: Time To Live degeri *ttl* kadar olan IP paketlerini kapsar.
ipversion <***ver***>: IP versiyonu *ver* olan IP paketlerini kapsar (ipv4, ipv6)
layer2 Sadece Layer 2 paketleri kapsar. Bu bir **IPFW2** opsiyonudur. IPFW1 ile çalismaz.
mac <***dst-mac***> <***src-mac***>: Hedef mac adresi *dst-mac* ve kaynal mac adresi *src-mac* olan paketleri kapsar. Bu bir **IPFW2** opsiyonudur. IPFW1 ile çalismaz.
setup: SYN bit'ini içerip ACK bit'ini içermeyen TCP paketlerini kapsar. Bu opsiyon "tcpflags syn, !ack" opsiyonunun kisaltilmis halidir.
tcpack: Sadece ACK bit'ini içeren TCP paketlerini kapsar.
tcpoptions <***spec***> TCP header'inda *spec* ile belirtilmis özellikleri tasiyan paketleri kapsar. Desteklenen TCP opsiyonlari asagida listelenmistir.

mss	maximum segment size
window	tcp window advertisement
sack	selective ack
ts	time stamp (rfc1323)
cc	t/tcp connection count (rfc1644)

---

* TRAFIK DÜZENLEME KONFIGÜRASYONU (DUMMYNET)

ipfw, paket filtrelemenin yaninda ***dummynet*** araciligi ile trafik düzenlemesi de yapabilmektedir.
Trafik düzenleyicisi, belirlenen özellikteki paketleri ayirir ve ayni statüde olan paketler için *pipe* denilen sanal bir baglanti açar.
pipe, verilen bantgenisligi, paket bekletme süresi, kuruk boyutu ve paket kayip oranlarinda sanal bir baglanti yaratir. Paket geçerken, ait oldugu *pipe*'in kurallarina göre geçer.
pipe konfigürasyonu asagidaki düzende uygulanir:

```
ipfw pipe <pipe_no> config <pipe konfigürasyonu>
```

<pipe konfigürasyonu:>

bw <*bandwidth*>:

Bantgenisligi [K|M] {bit/s | Byte/s } ile ölçülür.
Default degeri 0 dir. 0 limitsiz bantgenisligi anlamindadir. Bantgenisliginin miktari ile birimi arasinda bosluk birakilmamalidir.

```
ipfw pipe 1 config bw 1Mbit/s
```

delay *ms-delay*:

Paket bekletme süresi milisaniye ile ölçülür. Standart degeri 0 ms dir. Yani bekletme yoktur.

plr *kayip_orani*:

PLR=Packet Loss Rate, paket kayip orani. *kayip_orani* 0 ile 1 arasinda bir degerdir. 0 kayip yok, 1 ise %100 kayip anlamina gelir.
Asagidaki örnekte pipe 1'den geçen paketlerin %50si kaybolacaktir.

```
ipfw pipe 1 config plr 0.50
```

* ÖRNEKLER

ipfw'nin çok fazla kullanim sekli var. Bunlarin birkaçi açiklamalari ile birlikte asagida örneklenmistir:

BASIT PAKET FILTRELEME:

Asagidaki komut mevcut kurallari hit sayilari ile birlikte listelemer:

```
ipfw -a list
```

Asagidaki komut mevcut kurallari, hit sayilari ve son eslesme zamani ile birlikte listeler:

```
ipfw -at list
```

Asagidaki örnek, *www.kaynak.com* dan gelen *www.hedef.com* un *SSH* portuna giden TCP paketlerin geçmesini engeller.

```
ipfw add deny tcp from www.kaynak.com to www.hedef.com 22
```

Bu örnek, *123.45.0.0/16* dan *my.host.org*a gönderilen tüm paketlerinin geçmesini engeller. (123.45.0.0/16 = 123.45.0.0:255.255.0.0)

```
ipfw add deny all from 123.45.0.0/16 to my.host.org
```

TRAFIK DÜZENLEME:

Asagidaki örnek gelen paketlerin %10'unu siler.

```
ipfw add prob 0.10 deny all from any to any in
```

Siradaki örnek, yukarda yapilan islemi *dummynet* araciligi ile pipe kullanarak yapmaktadir

```
ipfw add pipe 2 all from any to any
ipfw pipe 2 config plr 0.10
```

Ag geçidi olarak kullanilan bir firewallda pipe'lari kullanarak bantgenisligi yapilabilir. Yerel agdaki 10.0.1.0/24 iplerinin trafigini limitleyelim.

```
ipfw add pipe 1 all from 10.0.1.0/24 to any out
ipfw pipe 1 config bw 300Kbit/s queue 50KBytes
```

Mevcut bantgenisliginin giris ve çikis olarak esit bir biçimde bölüstürmesi de giren ve çikan paketler için ayri kullar yazilmasiyla saglanir. (Bantgenisligi toplam 512Kbit/s kabul ediliyor.)

```
ipfw add pipe 1 all from any to any out
ipfw add pipe 2 all from any to any in
ipfw pipe 1 config bw 256Kbit/s
ipfw pipe 2 config bw 256Kbit/s
```

Trafik Düzenlemenin bir diger özelligi de paket bekletme asagidaki örnek disari çikan paketleri 250ms bekletmekte ve 512Kbit/s bantgenisligi limiti getirmektedir.

```
ipfw add pipe 1 all from any to any out
ipfw pipe 1 config delay 250ms bw 512Kbit/s
```

Beni FreeBSD dünyasina kazandiran Sn. Kutluhan KIBRIT'e sonsuz saygi ve tesekkürlerimi sunarim. Bu dökümantasyon, yazarin ismi ve ünvani degistirilmedigi sürece, kaynak gösterilmek sartiyla kullanilabilir.

Hazirlayan:

Özkan KIRIK
ozkan@mersin.edu.tr
Mersin Universty
Computer Research and
Application Center
System Administrator