Linux ile Ağ Servisleri

Hüsnü Demir
hdemir at metu.edu.tr
Ağ Destek Grubu
ODTÜ

Özet

Bu sunumda, bir pc sunucu üzerinde kurulacak olan Linux işletim sistemi, iptables, quagga (zebra), snort, netfilter ile temel ağ servislerinin verilmesi anlatılacaktır. Burada kurulumun adımları ve kurulacak olan sistemin yeteneklerini açıklamaya çalışmamıza karşın tüm adımlar verilmeyecektir.

Giriş

Ağ kurulumlarında eskiden beri ağ cihazları üreticilerinin ürettiği sistemler kullanılagelmiştir. Son yıllarda özgür yazılımların çoğalması ile uygulamalar artmış ve işe özel çözümler üretilmeye başlanmıştır. Bu çözümlerin içerisinde işletim sistemi olarak Linux ve BSD tabanlı sistemler öne çıkmıştır. Özgür yazılımlar aracılığı ile oluşturulan sistemler kullanım kolaylığı sağlamakla berarber bilgisayar teknolojilerinin ilerlemesi ile performans sorunları da azalmıştır. Özellikle küçük ve orta ölçekli ağlarda bu tür çözümlerin uygulanması yaygınlaşmıştır. Burada önemle üzerinde durulması gereken bir nokta bu sistemlerin iyi bir şekilde yönetilmesidir.

İşletim Sistemi

İşletim sistemi seçimi daha çok kullanıcının yada uygulayıcının tercihine bağlıdır. En iyi işletim sistemi üzerinde yeterli hakimiyetin sağlanabildiği bir işletim sistemidir. Bu açından baktığımızda pekçok alternatif çıksa da zamanla performans, uygulanabilirlik, güncelleme, kullanılabilirlik gibi ölçekler artmakta ve karşımıza Linux ve BSD tabanlı makineler çıkmaktadır.

Uygulamalarımızda pekçok defa BSD tabanlı (özellike FreeBSD) işletim sistemi kullanmış olmamıza rağmen Linux çekirdeğinin sahip olduğu Netfilter özelliklerinden faydalanmak amacı ile bu işletim sistemi seçilmiştir. Yaptığımız uygulamalarda (bizim açımızdan) Linux tabanlı, Debian sürümü, bir işletim sistemi ile BSD tabanlı, FreeBSD, arasında çok büyük farklılık olmadığını gördük. En son BSD tabanlı bir yönlendiricinin çalışma zamanı 460 gün olarak kaydedilmiştir. Gene aynı şekilde başka bir işlem için kullandığımız cihazın çalışma zamanı 500 gündür. Genelde bu sistemleri yer değiştirmeler ve elektrik sorunları yüzünden kapattığımızı da belirtmek isterim. Köprü olarak uzunca bir süre kullandığımız Debian cihazının çalışma zamanının (emekliye ayrılmadan önce) ~410 gün olduğunu belirtmek isterim.

İşletim sistemlerinde önemli olan konu kararlı bir durumu yakaladıktan sonra işletim sistemini güncellememektir. Bu konunun yanlış anlaşılmaması yerindedir. Eğer işletim sistemi çekirdeğinde sizi etkileyen bir sorun yok ise (güvenlik sorunu gibi) işletim sistemini olduğu gibi bırakmak gereklidir. Mesela; SSL uygulaması çalıştırmadığınız bir cihazda çıkan SSL güvenlik açığı için cihazınızı güncellemeniz çok gerekli değildir. Dolayısı ile bu tür sistemlerde otomatik güncelleme özerilmemektedir.

Hizmet verdiğiniz alanların sorun listelerine (güvenlik açıkları için özellikle) üye olmanız gereklidir. Buun yanında geliştirme listeleri, kullanıcı listeleri gibi çeşitli listelere üye olunmalıdır. Web sayfalarında güvenlik başta olmak üzere tüm gelişmelere devamlı takip edilmelidir. Bazı önemli siteler, http://www.cisecurity.com, http://www.nsa.gov, http://www.microsoft.com/security, http://www.sans.org, http://www.eeye.com, http://www.securityfocus.com, http://www.cert.org ve http://csirt.ulakbim.gov.tr.

Kurulum

İşletim sistemi kurulumundan önce yapılacak en önemli iş kurulum yapılacak cihazın özelliklerinin belirtildiği bir tablo olacaktır. Bu işlem hem kurulum esnasında hem de daha sonradan çıkabilecek sorunlarda işe yarayacaktır. Bu listede CPU, slotları ve tipleri ile beraber memory (bellek), disk kapasitesi ve tipi, eğer varsa SCSI kartı tipi, elektrik gereksinimleri, üzerine takılı ekstra cihazlar ile ilgili bilgiler ve son olarak ethernet kartları ile ilgili bilgiler olmalıdır. Bu bilgilere cihazın slot sayısı, bu slotların tipleri, hızları ve yerleri de eklenmelidir. Ethernet kartı yüksek kapasite de çalışacak cihazlar için önemlidir. Bu kartların chipsetlerinin özellikle işletim sistemi için araştırılması ve mümkünse performans testlerinin yapılması gereklidir.

Diğer önemli nokta ise her ağ uzmanının atladığı bir nokta olan işletim sistemi kurulum notlarını okumak ve yayınlanmış bültenleri takip etmektir. Hiç olmaz ise bu notlar şöyle bir gözden geçirilmelidir. Tabii ilk defa bu işe girişen yöneticiler bu notları tamamen okumalıdır.

Standart olarak işletim sisteminin sadece base denilen kısmın kurulması ve daha sonra ihtiyaç oldukça diğer parçaların eklenmesi hem disk kapasitesi ihtiyacını düşürür, hemde işletim sistemi üzerinde bulunan yazılımlarda çıkan güvenlik açıkları ile uğraşmak daha kolay olacaktır. Debian'ın son sürüm olarak yakın zamanda çıkacak *etch* kod adlı sürümü bu işlevi sağlamaktır. Günümüz bilgisayar sistemlerinde CD'den yükleme yapılabilmekle beraber başka alternatiflerde kullanılabilir (bkz http://www.debian.org). Kurulum esnasında her dağıtım gibi Debian'da standart bir çekirdek (kernel) kurar. Bu çekirdek çok genel bir çekirdek olup daha sonra rahatlıkla baştan derlenebilmektedir.

Kurulum esnasında sorulan sorunlardan (eğer sorun çıkmaz ise) en önemlisi disk ile ilgili olanıdır. Disk yapılandırması önemlidir. Eğer tüm loglama işlemi cihaz üzerine yapılacak ise sistemin hızlı bir depolama sistemine sahip olması tercih edilir. Disk bölümlemesi buna uygun yapılmalıdır.

Kurulum esnasında ağ bağlantısı sağlanırsa, kurulum ağdan da yapılabilir. Böylece güvenlik güncellemelerinin de anında yapılması sağlanır. Sistem son olarak şifre soracaktır. Bu şifrenin en az 10 karakterli, sağlam bir yapıda olmasını sağlamanızı öneririz.

İşletim sisteminin tek seferde kurulduğunu varsaysak bile pek çok sistem için böyle olmadığını belirtmek isterim. Çıkan pekçok sorunun başkaları tarafından daha önce yaşandığını ve çözüm üretildiğini unutmamak lazım. Bundan dolayı her türlü sorun da ilk olarak http://www.google.com.tr adresine, daha sonra ise listelere (http://forum.ulak.net.tr gibi) başvurmak sorunların çözümünde yararlı olacaktır.

Temel kurulum yaptıktan sonra cihazın İnternet'e bağlanması işlerimizi kolaylaştıracaktır. İnternet'e bağlı her cihazın mutlaka güvenlik önlemlerini almak gereklidir. "*aptitude update*" ve *"uptitude upgrade"* işlemleri ile gerekli paketlerin güncellenmeleri kolaylıkla sağlanır. Daha sonra iptables ile güvenlik güçlendirilebilir.

```
iptables –A INPUT –i lo –j ACCEPT
iptables –A INPUT  –m state --state INVALID –j DROP
iptables –A INPUT –m state --state RELATED,ESTABLISHED –j ACCEPT
```

iptables –A INPUT –j DROP
iptables –A OUTPUT –j ACCEPT

Şimdi sırada kişiselleştirilmiş bir çekirdeğin derlenmesine geldi. Daha önce hazırlamış olduğumuz bilgisayar aksesuar listesi ve hangi özellikleri isteyeceğimiz bilmemiz bizim işimizi kolaylaştıracaktır.

Çekirdek derlemede ilk önce kaynak dosyalarını almamız lazım. Bunun için en iyi yöntem ftp://ftp.tr.kernel.org adresinden kernel dosyasını elde etmektir.

```
# cd /usr/src/
# aptitude install ncftp
# ncftp ftp.tr.kernel.org
NcFTP 3.2.0 (Aug 05, 2006) by Mike Gleason (http://www.NcFTP.com/contact/).
Connecting to 144.122.144.146...
Middle East Technical University * Anonymous * FTP Server
Logging in...
Anonymous access granted, restrictions apply.
Logged in to ftp.tr.kernel.org.
ncftp / > cd pub
ncftp /pub > cd linux/
ncftp /pub/linux > cd kernel/
ncftp /pub/linux/kernel > cd v2.6
ncftp /pub/linux/kernel/v2.6 > ls -l L*
-r--r--r--   1 ftp     ftp         0   Nov 29 22:11   LATEST-IS-2.6.19.2
ncftp /pub/linux/kernel/v2.6 > get linux-2.6.19.2.tar.bz2*
linux-2.6.19.2.tar.bz2:                    40.75 MB  256.03 kB/s
linux-2.6.19.2.tar.bz2.sign:                 248.00 B   25.86 kB/s
ncftp /pub/linux/kernel/v2.6 > bye
# gpg --keyserver wwwkeys.pgp.net --recv-keys 0x517D0F0E
# gpg --verify linux-2.6.19.2.tar.bz2.sign linux-2.6.19.2.tar.bz2
gpg: Signature made Wed 10 Jan 2007 05:21:53 PM EST using DSA key ID 517D0F0E
gpg: Good signature from "Linux Kernel Archives Verification Key <ftpadmin@kernel.org>"
Primary key fingerprint: C75D C40A 11D7 AF88 9981  ED5B C86B A06A 517D 0F0E
# tar jxf linux-2.6.19.2.tar.bz2
# ln -s linux-2.6.19.2 linux
# cd linux
# less README
# make mrproper

Eğer eski konfigüraston dosyanız varsa .config olarak bunu kullanabilirsiniz.
# make oldconfig

Veya yeniden başlamak için;
# make menuconfig

Genelde kullanma ihtimalimiz olan ama şu an için kullanmayacağız kısımları modül olarak işaretlemede fayda vardır.
Şunlar işaretli olsa iyi olur:
Code maturity level options  ---> [*] Prompt for development and/or incomplete code/drivers
Networking  ---> Networking options  --->
        [*] Network packet filtering (replaces ipchains)  --->
                Core Netfilter Configuration  --->
                IP: Netfilter Configuration  --->
                        Tamamını Modül olarak işaretleyebiliriz. Ipv6 ile ilgili bir şey yapmayacağız. Kullanılmayan herşeyi
                        geçersiz kılmakta yarar var.
        QoS and/or fair queueing  --->
                Tamamını Modül olarak işaretleyebiliriz.
Device Drivers  ---> Network device support  --->
        Ilgili kartları seçebilirsiniz.
```

Diğer seçenekleri help opsiyonu aracılığı ile tanıyıp seçebilirsiniz. Genelde önceden seçilmiş olanlar kalsa da pekçoğunun gereksiz olduğunu görebilirsiniz. Burada unutulmaması gereken nokta disk sürücülerinin modül değil çekirdek içerisinde derlenmiş olması gerekir. Seçimleri ilgili donanıma göre yapılandırdıktan sonra *"make", "make install"* ve *"make modules_install"* yapılmalıdır. Minumum kurulum gerçekleştirdiğimiz için bazı paketlerin kurulmasına ihtiyaç duyarsa bunları aptitute paket yönetim yazılımı ile kolaylıkla yükleyebilirsiniz. Eğer **grub** kullanıyorsanız *"update-grub"* komutunu kullanmanız gerekecektir. Gerekli

bilgiler için http://www.debian.org , http://kerneltrap.org ve http://www.kernel.org adreslerinden yararlanılabilir.

Burada netfilter modülünü özelleştirmek mümkündür. Bunun için http://www.netfilter.org adresi çok önemli bir kaynaktır. *"aptitute install iptables"* ile gerekli paket kurulabilir. Bu sitede bulunan özellikle **ipset** paketi çok işlevsel olabilmektedir.

Burada p2p yönetimi için ip2p ve L7-filter çok güzel bir kaynaktır. QoS ile beraber çok işlevsellikler kazandırılabilir.

Log Tutma

Log tutmak için **fprobe** uygulaması kullanılacaktır. Bu uygulama ağ izlerini cisco netflow verisi şeklinde log tutma makinasına göndermektedir. Benzer şekilde fprobe-ng ve fprobe-ulog'da aynı işleri yapan daha yetkin uygulamalardır. *"aptitude install fprobe-ng"* diyerek kurulumu gerçekleştirebiliriz. *"/etc/default/fprobe"* dosyası içerisindeki INTERFACE ve FLOW_COLLECTOR parametreleri kullanılarak gerekli ayarlamalar yapılır. Bu günlükleri toplamak için flow-tools kullanılabilir.

Zebra veya Quagga

Zebra Quagga uygulamasının ve daha pekçok uygulamanın atası şeklindedir. Zaman zaman güncellemeleri yapılsa da şu anda daha aktif olan Quagga yönlendirme yönetim aracıdır. Quagga pek çok bölümden oluşmaktadır. Bunlardan bizlerin genelde kullandığı **zebra'**dır. Özellikle büyük ağlarda **ospfd, ripd** uygulamaları da kullanılmaktadır. Sınır yönlendiriciler başta olmak üzere çoklu bağlantılar için **bgpd** uygulaması da kurulabilir.

```
#aptitute install quagga
```

Veya

```
# wget http://www.quagga.net/download/quagga-0.99.6.tar.gz
# tar zxvf quagga-0.99.6.tar.gz
# cd quagga-0.99.6
# ./configure --disable-ipv6  --disable-bgpd --disable-ripd  --disable-ripngd
 # make
 # make install
# cd /usr/local/etc/

Başlangıç için;
# cp zebra.conf.sample zebra.conf
# less /usr/local/etc/zebra.conf
!
! Zebra configuration saved from vty
!   2007/04/06 10:06:34
!
hostname Router
password zebra
enable password zebra
log file /var/log/zebra.log
!
interface eth0
!
interface eth1
!
interface lo
!
access-list NOLOGIN permit 127.0.0.1/32
access-list NOLOGIN deny any
!
!
line vty
```

```
 access-class NOLOGIN
!
# cp ospfd.conf.sample ospfd.conf
# less /usr/local/etc/ospfd.conf
!
! Zebra configuration saved from vty
!   2006/09/29 14:43:43
!
hostname Router-ospfd
password zebra
enable password zebra
log file /var/log/ospfd.log
!
interface eth0
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 ospf-key
 ip ospf hello-interval 30
 ip ospf dead-interval 120
!
interface eth1
!
interface lo
!
router ospf
passive-interface eth1
 network 10.1.0.0/24 area 0.0.0.0
!
access-list NOLOGIN permit 127.0.0.1/32
access-list NOLOGIN deny any
!
line vty
 access-class NOLOGIN
!
```

Sistemi çalıştıralım. –P opsiyonu ile varsayılan port dışında bir portta çalışmasını sağlamak güvenlik için yararlı olabilir.

```
# /usr/local/sbin/zebra –d –P 4000
# /usr/local/sbin/ospfd –d –P 4001
```

Kurulum esnasında sorulan sorular yönetim ile ilgili olup genelde yukarıda bahsettiğim iki bölüm ve eğer isteniyorsa SNMP desteği kurulabilir. İstatistiksel bilgiler toplayıp bu bilgileri rrdtool (http://oss.oetiker.ch/rrdtool) ile görüntülemek çoğu zaman yararlı olmaktadır.

Quagga sisteminin kullanımı **Cisco** kullanımına çok benzemektedir. Dolayısı ile bu konuda özellikle İnternet'ten pekçok bilgi edinilebilinir. Bunların pekçoğu çalışacaktır.

Snort

Yönlendirici cihazlarda sniffer türevi yazılımların bulunması pekçok yönden yarar sağlamaktadır. Ama unutulmamalıdır ki bu sistemlerin ele geçirilmesi çok büyük sorunlara neden olacaktır. Bu yüzden güvenliği en üst seviye de tutmak gerekmektedir.

Snort uygulamasını kurmak ve işletmek için pekçok yöntem önerilmektedir. Snort sistemi IDS olarak çalışmakla beraber IPS olarak **INLINE** özelliği ile de kullanılabilmektedir. IPS özelliğini kullanmak için güvenlik duvarı kurulum yöntemlerinden *"herşeyi engelle sadece gerekli olanlara izin ver"* yöntemini kabul etmek ve ona göre bir strateji uygulamak gereklidir. IPS yöntemi için pekçok kuraldan sadece gerekli olanlarını seçmek ve ağ özelliklerine göre, özellikle ağ kullanım politikasına göre, gerekli kural listesi tanımlanmalıdır. Bu ise küçük ağlarda uygulanabilirliği yüksek, ama yüksek kapasiteli bağlantılar için sorunlu olmaktadır. Bu yüzden IPS sistemlerinin arada olması istenmeyen bir durum olmaktadır. Tabii her zaman için bu para/politika paradoksu işerisinde bir kavramdır. Eğer yapılabiliyorsa niye yapılamasın.

Snort uygulamasını tekrar aptitute ile kurmak kolay olmakla beraber, IDS/IPS sistemlerinde son kararlı sürüm kullanmanın önemi büyüktür. Bu yüzden ilk önce http://snort.org sitesine üye olmak ve kaynak programları buradan edinmek lazımdır.

 # wget http://www.snort.org/dl/current/snort-2.6.1.2.tar.gz
 # wget http://www.snort.org/dl/current/snort-2.6.1.2.tar.gz.md5
 # md5sum snort-2.6.1.2.tar.gz
22c448e25538cdf74c62abe586aeac0a snort-2.6.1.2.tar.gz
 # cat snort-2.6.1.2.tar.gz.md5
22c448e25538cdf74c62abe586aeac0a snort-2.6.1.2.tar.gz
#

Daha sonra kural dosyalarını indirmek gerekiyor. Snort.org sitesinden kural dosyalarını indirebilmek için REGISTER olmanız gerekmektedir.
 # wget http://www.snort.org/pub-bin/downloads.cgi/Download/vrt_os/snortrules-snapshot-CURRENT.tar.gz
 # wget http://www.snort.org/pub-bin/downloads.cgi/Download/vrt_os/snortrules-snapshot-CURRENT.tar.gz.md5
 # wget http://www.snort.org/pub-bin/downloads.cgi/Download/comm_rules/Community-Rules-CURRENT.tar.gz
 # wget http://www.snort.org/pub-bin/downloads.cgi/Download/comm_rules/Community-Rules-CURRENT.tar.gz.md5
 # wget http://www.bleedingsnort.com/bleeding.rules.tar.gz

 # tar zxf snort-2.6.1.2.tar.gz
 # cd snort-2.6.1.2
 # less RELEASE.NOTES
 # less doc/INSTALL
 # ./configure --enable-dynamicplugin --enable-pthread
 # make
 # make install

Sistemi test etmek için;

```
# snort -evi eth0
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Var 'any_ADDRESS' defined, value len = 15 chars, value = 0.0.0.0/0.0.0.0
Var 'lo_ADDRESS' defined, value len = 19 chars, value = 127.0.0.0/255.0.0.0
Verifying Preprocessor Configurations!

Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.6.1.2 (Build 34)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2006 Sourcefire Inc., et al.

Not Using PCAP_FRAMES
01/22-03:30:58.187791 0:18:FE:78:2D:7C -> 0:11:85:C0:D7:97 type:0x800 len:0x86
144.122.3.141:22 -> 144.122.3.228:1046 TCP TTL:64 TOS:0x10 ID:34623 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0xD0C5F09D  Ack: 0x5BA57B6E  Win: 0xF53C  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Sistemi kullanmadan önce snort.conf dosyasını ayarlamak gerekmektedir;
 # vi snort.conf
var HOME_NET [10.1.1.0/24,192.168.1.0/24]

Genelde varsayılan tanımlar başlangıç için yeterli olmaktadır. Daha iyi bir ayarlama için tavsiyemiz her zamanki gibi kullanım klavuzunu (http://www.snort.org/docs/) okumak olacaktır;
Günlüğü nasıl tutacağınıza karar vermeniz gerekecektir. Eskiden beri kullanılan en kolay yollardan biri;
output alert_fast: snort-alert-fast.log
Veya;
output database: log, mysql, user=root password=test dbname=db host=localhost
Tabii veritabani uygulaması için en başta uygulamayı derlerken (*--with-mysql*) opsiyonunu kullanmak gerekir. Daha sonra MYSQL kurulu cihaz üzerinde;
 # mysql SNORT -p < ./schemas/create_mysql

Komutu uygulanır. Bu komut sayesinde snort'un kullanacağı tablolar oluşturulur. "*create_mysql****" ve diğer dosyalar snort'un kaynak dosyalarının içerisindedir. Oluşturulan snort.conf dosyasındaki iz dosyaları irdelenmeli ve daha önce de indirdiğimiz dosyalarda göz önünde bulundurulmalıdır.***

 # snort -T -c /usr/local/etc/snort/snort.conf

Komutu ile de oluşturulan dosyanın test edilmesi sağlanır. Son olarak;

 # snort -Dd -c /usr/local/etc/snort/snort.conf

Komutu ile snort'un arka planda çalışması sağlanır. Günlük bilgileri "*/var/log/snort****" görmeye başlayabilirsiniz. Tabii eğer veritabanı opsiyonu kullanmadı iseniz.***

Burada önemli olan nokta kurulumun tüm olarak anlatılmadığı ve pek çok ara basamağın atlandığıdır. Mesela "*/var/log/snort*" kütüğü olmadığı için alınacak hata mesajı için bu kütük oluşturulmalıdır (*mkdir –p /var/log/snort*). Aynı şekilde kural dizini olmadığı hatasını aldığınız zaman;

```
# mkdir –p /usr/local/snort/rules
# tar zxf snortrules-snapshot-CURRENT.tar.gz
# tar zxf Community-Rules-CURRENT.tar.gz
# tar zxf bleeding.rules.tar.gz
# cp rules/* /usr/local/snort/rules/
```

Böylece tüm kural silsilesi tek bir kütük içerisine yerleştirilmiş olacaktır. Kuralların otomatik güncellemesi için http://oinkmaster.sourceforge.net/ adresinden faydalanabilirsiniz.

RRDTOOL ve LOG Tutma

RRDTool çok işlevsel bir log tutma sistemidir. Ayrıca tutlan bu loglar ile grafikler oluşturabilmektedir. Debian sistemine "aptitute install rrdtool" ile kolaylıkla yüklenebilmektedir. Genelde standart yöntemler daha çabuk ve temiz bir yükleme sağlamaktadır. İlk önce veritabanı dosyalarını oluşturalım.

```
#!/bin/sh
cd /var/log/rrdtool/
if [ ! -f bandwith_eth0.rrd ]; then
rrdtool create bandwith_eth0.rrd --start N \
      DS:inByte:COUNTER:600:U:U \
      DS:outByte:COUNTER:600:U:U \
      DS:inPKT:COUNTER:600:U:U \
      DS:outPKT:COUNTER:600:U:U \
      DS:inDrop:COUNTER:600:U:U \
      DS:outDrop:COUNTER:600:U:U \
      RRA:AVERAGE:0.5:1:105120 RRA:MAX:0.5:6:35040
fi

if [ ! -f bandwith_eth1.rrd ]; then
rrdtool create bandwith_eth1.rrd --start N \
      DS:inByte:COUNTER:600:U:U \
      DS:outByte:COUNTER:600:U:U \
      DS:inPKT:COUNTER:600:U:U \
      DS:outPKT:COUNTER:600:U:U \
      DS:inDrop:COUNTER:600:U:U \
      DS:outDrop:COUNTER:600:U:U \
      RRA:AVERAGE:0.5:1:105120 RRA:MAX:0.5:6:35040
fi

if [ ! -f meminfo.rrd ]; then
rrdtool create meminfo.rrd --step 300 --start N \
            DS:MemTotal:GAUGE:600:0:U \
            DS:MemFree:GAUGE:600:0:U \
            DS:Buffers:GAUGE:600:0:U \
            DS:Cached:GAUGE:600:U:U \
            DS:SwapCached:GAUGE:600:U:U \
            DS:Active:GAUGE:600:U:U \
            DS:Inactive:GAUGE:600:U:U \
            DS:HighTotal:GAUGE:600:U:U \
            DS:HighFree:GAUGE:600:U:U \
            DS:LowTotal:GAUGE:600:U:U \
            DS:LowFree:GAUGE:600:U:U \
```

```
        DS:SwapTotal:GAUGE:600:U:U \
        DS:SwapFree:GAUGE:600:0:U \
        DS:Dirty:GAUGE:600:0:100 \
        DS:Writeback:GAUGE:600:0:100 \
        DS:Mapped:GAUGE:600:0:100 \
        DS:Slab:GAUGE:600:0:100 \
        DS:CommitLimit:GAUGE:600:0:100 \
        DS:Committed_AS:GAUGE:600:0:100 \
        DS:PageTables:GAUGE:600:0:100 \
        DS:VmallocTotal:GAUGE:600:0:100 \
        DS:VmallocUsed:GAUGE:600:0:100 \
        DS:VmallocChunk:GAUGE:600:0:100 \
        RRA:AVERAGE:0.5:2:2000

fi

if [ ! -f loadavg.rrd ]; then
rrdtool create loadavg.rrd --step 60 --start N \
        DS:Load1Min:GAUGE:60:0:U \
        DS:Load5Min:GAUGE:60:0:U \
        DS:Load15Min:GAUGE:60:0:U \
        RRA:AVERAGE:0.5:1:86400
fi
cd –
```

Arayüzlerin istatistiklerini elde etmek için;

```
#!/usr/bin/perl -w

$RRDUPDATE="/usr/bin/rrdupdate";
$DBPATH="/var/log/rrdtool";

$values=(qx!/bin/cat /proc/net/dev!);
@values=split(/\n/,$values);

# Eth0 icin;
@interface_eth0=split(/\W+/,$values[3]);
$inByte="$interface_eth0[2]";
$inPKT="$interface_eth0[3]";
$inDrop="$interface_eth0[5]";
$outByte="$interface_eth0[10]";
$outPKT="$interface_eth0[11]";
$outDrop="$interface_eth0[12]";
$tmp=(qx!$RRDUPDATE $DBPATH/bandwith_eth0.rrd N:$inByte:$outByte:$inPKT:$outPKT:$inDrop:$outDrop
!);

# Eth1 icin;
@interface_eth1=split(/\W+/,$values[4]);
$inByte="$interface_eth1[2]";
$inPKT="$interface_eth1[3]";
$inDrop="$interface_eth1[5]";
$outByte="$interface_eth1[10]";
$outPKT="$interface_eth1[11]";
$outDrop="$interface_eth1[12]";
$tmp=(qx!$RRDUPDATE $DBPATH/bandwith_eth1.rrd N:$inByte:$outByte:$inPKT:$outPKT:$inDrop:$outDrop
!);
```

CPU Yük durumu için;

```
#!/usr/bin/perl  -w
$RRDUPDATE="/usr/bin/rrdupdate";
$DBPATH="/var/log/rrdtool";
$values=(qx!/bin/cat /proc/loadavg!);
@value=split(/\ +/,$values);
$putit="N:" . $value[0] . ":" . $value[1] . ":" . $value[2];
$tmp=(qx!$RRDUPDATE $DBPATH/loadavg.rrd $putit !);
```

Hafıza bilgileri ile ilgili genel istatistiksel bilgiler için;

```
#!/usr/bin/perl -w
$RRDUPDATE="/usr/bin/rrdupdate";
$DBPATH="/var/log/rrdtool";
$values=(qx!/bin/cat /proc/meminfo!);
@values=split(/\n/,$values);
$putit = "N" ;
foreach (@values) {
    @value=split(/\W+/,$_);
    $putit=$putit . ":" . "$value[1]";
}
$tmp=(qx!$RRDUPDATE $DBPATH/meminfo.rrd $putit !);
```

Grafik çizmek için ise şöyle bir örnek yeterli olacaktır;

```
#!/bin/sh

RRDTOOL="/usr/bin/rrdtool graph "
GRAPHS="/var/log/rrdtool/graph/new"

date=`date +"%F %R"`
cd /var/log/rrdtool/

$RRDTOOL $GRAPHS/kampus-day.png --start -1d -w 600 -h 150 -v bps \
    "DEF:in=bandwith_eth0.rrd:inByte:AVERAGE" "DEF:out=bandwith_eth0.rrd:outByte:AVERAGE" \
    'CDEF:bin=in,8,*'    'CDEF:bout=out,8,*' \
    'AREA:bout#00CC00:Gelen Trafik   ' 'LINE2:bin#000066:Giden Trafik:STACK \l' \
    "COMMENT:\n" \
    'GPRINT:bin:LAST:Anlik Giden \:%3.2lf%sbps ' 'GPRINT:bin:AVERAGE:Ortalama Giden \:%3.2lf%sbps ' 'GPRINT:bin:MAX:Maksimum
Giden \:%3.2lf%sbps \l' \
    'GPRINT:bout:LAST:Anlik Gelen \:%3.2lf%sbps' 'GPRINT:bout:AVERAGE:Ortalama Gelen \:%3.2lf%sbps'
'GPRINT:bout:MAX:Maksimum Gelen \:%3.2lf%sbps \l' \
    "COMMENT:\n" \
    "COMMENT: Güncelleme : $date \r"
```

"**U**" işareti ingilizce "**Unlimited**" kısmından geliyor. Buraya eğer değerleri biliyorsanız doğrudan yazabilirsiniz. Aynı zamanda bazen grafiklerde büyük atlamalar olmaktadır. Bunu düzeltmek için "**removespikes.pl**" betiği kullanılabilir. BU betik *google* aracılığı ile kolaylıkla elde edilebilir.

Bu konuda son bir uyarı da RRDTool versiyonları arasındaki farklılıklardır. Bu yüzden yukarıdaki betikler sizde tam olarak çalışmayabilir ama çalışabilir vaziyete getirilebilir.

İnce Ayarlar

Cihazı devreye almadan önce bazı ayarlamalar yapılabilir. İşletim sistemi bazında yapılacak ilk şey eğer daha önce yapılmadı ise yönlendirmeyi aktif hale getirmektir.

```
# sysctl -w net.ipv4.ip_forward=1
```

Eğer birden fazla bağlantınız var ise (*net.ipv4.conf.all.rp_filter=0*) olmasına dikkat etmenizi öneririm. Aynı zamanda "**/etc/sysctl.conf**" ayarları;

```
# cat sysctl.conf
net/ipv4/ip_conntrack_max=1310720
net/ipv4/tcp_sack=0
net/ipv4/tcp_timestamps=0
net/core/rmem_max=16777216
net/core/wmem_max=16777216
net/ipv4/tcp_rmem = 4096 87380 16777216
net/ipv4/tcp_wmem = 4096 65536 16777216
net/core/netdev_max_backlog=300000
net/ipv4/tcp_fin_timeout=15
net/ipv4/netfilter/ip_conntrack_tcp_timeout_fin_wait=15
net/ipv4/conf/default/accept_source_route=0
net/ipv4/conf/default/accept_redirects=0
net/ipv4/conf/all/accept_source_route=0
net/ipv4/conf/all/accept_redirects=0
```

Netfilter'ın **conntrack** modülünün parametrelerine dikkat edin. Bu sayının çok fazla olması sorun çıkarabilmektedir. Aynı zamanda bu modülü yüklerken parametre olarak **hashsize** vererek performans arttırılabilir. **modprobe.conf** dosyasına aşağıdaki saıtır ekleyin.

```
options ip_conntrack hashsize=163840
```

Unutmayınki çok güzel bir özellik olmasına rağmen bağlantı izleme modülü (**conntrack**) performansı olumsuz yönde etkilemektedir. Bu ise, özellikle bağlantınız hızlı ise, sistemi UDP saldırılarına açık bir hale getirmektedir.

Bu ayarlar size bir başlangıç olabilir. Her sistemin kendine özgü bir yapısı vardır. Dolayısı ile başlangıçta ümitsizliğe kapılmadan deneme yanılma yöntemi ile size en uygun çözümü bulabilirsiniz. Bu esnada tabii kullanıcılarınızdan bazı şikayetler gelecektir.

Sonuç

Burada Debian işletim sistemi iptables, quagga (zebra), snort ve netfilter ile yapılandırılmış ve güçlü bir güvenlik duvarı (netfilter) ve IDS (snort) özelliği olan bir yönlendirici oluşturulmuştur. Bu sistem pekçok ağlar için yeterli olabilmektedir. Bu tür sistemlerin kurulumu ve işletilmesi zahmetli görülsede son zamanlarda çıkan paket yönetim arayüzleri ile bu işler kolaylaşmıştır. Bu tür güzenlik sistemlerinin kurulumu kolay olsa da gayet ciddi güvenlik tedbirleri ile işletilmesi gerekmektedir. Nitekim bu tür sistemlerin ele geçirilmesi ile ciddi güvenlik sorunları oluşmaktadır.