FreeBSD "Sunucu" Optimizasyonu ve Güvenlik Ayarları.

Onur BEKTAŞ

Ekim 2006

onur [at] ulakbim [nokta] gov [nokta]tr

İÇERİK

Giriş

Niye Sunucu lafını kullandık? Çünkü işletim sisteminden beklentileriniz onu masaüstü yada sunucu olarak kullanmanıza göre çok farklılık gösterir. Unix'lerin ilk kurulumda gelen ayarları, sistemin kararlı çalışması düşünülerek ortalama bir kullanıcı ve giriş seviye bir sunucu düşünülerek optimize edilmiştir. Eğer işletim sistemini sunucu ayarlarında kullanacaksanız buna göre bazı sistem değişkenlerini uygulamanıza göre optimize etmeniz gerekir.

Burada yazılan bilgiler FreeBSD spesifik olmasına rağmen aynı konular benzer değerlerle ve başka isimlerle diğer işletim sistemleri içinde geçerlidir. Konuları ayrıntılı olarak anlatmadım bunun iki sebebi var: İlki anlatsak kitap olur, ikincisi sunucu kuruyorsanız zaten belli bir seviyede kurulum bilgisine sahip olmanız gerektiğini farzetmem. Bu sebepten örneğin nasıl çekirdek derlenir türü bir kısım yazmadım.

Örnek sunucumuz iki adet Opteron 250 işlemci, 4 Gbyte RAM ve 100 Mbit ethernet bağlantısına sahip. DNS sunucusu olarak kullanılacağı için ayarlar bind yazılımına göre yapıldı.

Yazıda belirtilen değerler sizin sisteminize ve uygulamanıza göre değişiklik gösterebilir. En uygun değerleri birazda deneyerek bulmanız gerekecektir.

Başlamadan son olarak legal uyarımızı yapalım…

Burada anlattıklarımı uygularken sisteminizi patlatırsanız sorumlusu ben değilim :). Sonra yok efendim çekirdeğim derlerken patadı, sistemi yardım diye yakınmayın kullanmak sizin sorumluluğunuz.

Donanım Seçimi

Sunucu için donanım almadan önce uygulamanızın ne gibi bir donanımsal gereksinimi olduğuna karar vermeniz gerekir. Bazı uygulamalar yapısı gereği daha çok belleğe ihtiyaç duyarlar ve genel olarak fazla işlemci gereksinimleri yoktur (squid,bind..). Bazılarının yaptığı işin işlemci gücü gereksinimi vardır ama o kadar bellek istemezler (iptables,ipfilter...) ve bazıları da kritik olarak hızlı diskler ve belleğe ihtiyaç duyarlar (mysql,postgresql...).

Sabit bir bütçeniz olduğunda örneğin DNS sunucusu yapılandıracaksanız paranızı çok daha iyi bir işlemci almaktansa ortalama seviyede alıp bolca bellek almak daha mantıklıdır. Kuracağınız yazılımın donanım gereksinimleri ile ilgili belgeleri önceden okumak genelde yararlıdır. O zaman niye gidip 2 işlemcili makineye DNS kurulum yaptınız diyeceksiniz. DNS sunucusu normalde çok işlemci gücü istememesine rağmen cache cleaning denilen süresi dolmuş dns bellek kayıtlarının bellekten temizlenmesi esnasında anlık yüksek işlemci gereksinimi doğurabiliyor. DNS sunucumuzun tampon belleğinde 5 milyon civarında kayıt olduğu için cache cleaning esnasında normalde %10-12 olan CPU kullanımı %70-80 civarına çıkabiliyor.

Sistemin ilk kurulumumunda dikkat etmeniz gerekenler:

- Tek diskiniz varsa tampon bellek (“swap”) alanınızı diskin başından verin, Yani ilk yarattığınız partition tampon bellek olsun. Diskin silindirleri en diş taraftan en içe doğru sıralandığı için (ve diskin en dış kısmı en hızlı erişilen kısmı olduğu için) eğer tampon bellek ihtiyacınız doğarsa diskin en hızlı erişilen kısmında olması gerekir.
- Eğer iki diskiniz varsa tampon bellek alanını işletim sistemin koyduğunuz diske koymayıp ikinci diske tanımlayın. Alternatif olarak eğer işletim sisteminizi kurduğunuz diskte çok I/O olmayacağını düşünüyorsanız iki diskede swap alanı tanımlayıp performansı arttırabilirsiniz
- En az bellek miktarınızın iki katı kadar swap alanı ayırın.
- Kurulumda Custom->Distributions->Custom altında base,man,src,ports 'u seçin boş yere diğerleri yer kaplamasın.
- Diskleri label editorle paylaştırıken ile mutlaka soft update'leri aktive edin (Daha fazla bilgi için FreeBSD handbook).
- Sunucu sisteminize gereksiz hiç bir paket kurmayın. Örneğin çok özel bir sebebi yoksa XWindows kurmayın. Kuracağınız her fazladan paket sisteminizde güvenlik açığı çıkma olasılığını ve takip etmeniz gereken yama miktarını arttırır.

İlk Kurulumdan sonra yapılması gerekenler:

Derleme optimizasyonu.

Bundan sonra yapacağımız şeyler hep kod derlemek üzerine olacağı için ilk olarak /etc/make.conf dosyasını yaratarak derleme optimizasyon flag`larımızı yazalım. Bu dosyanın örneğini /usr/local/src/share/examples/etc altında bulabilirsiniz. Dosya içinde yer alan açıklamalara göre kendi sisteminize uyan değişiklikleri yapabilirsiniz. Ama yanlış birşey yazarsanız program derlemekte güçlük çıkabilir. Örneğin benim burda yazdığım -O3 opsiyonu sorun çıkaranlardan biridir, eğer derleme problemi yaşarsanız -O2 ile değiştirin. Burada değiştirilecek diğer bir kısımda CPUTYPE. Bizim işlemci Opteron olduğu için opteron yazdık ama işlemci tipinizde göre aşağıdakilerden birini seçip yazın:

```
# (AMD CPUs) opteron athlon64 athlon-mp athlon-xp athlon-4
# athlon-tbird athlon k8 k6-3 k6-2 k6 k5 nocona
# (Intel CPUs) pentium4[m] prescott pentium3[m] pentium2 penitumpro
# pentium-mmx pentium i486 i386
# Alpha/AXP architecture: ev67 ev6 pca56 ev56 ev5 ev45 ev4
# AMD64 architecture: opteron, athlon64, nocona
# Intel ia64 architecture: itanium2, itanium
```

Cat komutu ile (yada favori editörünüzle) /etc/make.conf dosyasını oluşturun. CTRL+C ile cat komutundan cıkabilirsiniz

```
[root@alchemist ~]# cat > /etc/make.conf
CPUTYPE=opteron
CFLAGS= -O3 -pipe -funroll-loops -ffast-math
COPTFLAGS= -O2 -pipe -funroll-loops -ffast-math
```

Cvsup ile kaynak kodlarının güncellenmesi.

Sırada cvsup ile port kolleksiyonunu ve kaynak kodlarını son haline getirmek var. Bunun için /etc/stable-supfile dosyası oluşturup içine aşağıdakileri yazın. Burada değiştirmeniz gereken satır default release satırı. FreeBSD sürümümüz 6.1 olduğu için 6.1 olarak yazıldı. Kurduğunuz sürüme göre bu satırı değiştirin:

```
[root@alchemist ~]#  cat > /etc/stable-supfile
*default host=cvsup.tr.FreeBSD.org
*default base=/usr
*default prefix=/usr
*default release=cvs tag=RELENG_6_1
*default delete use-rel-suffix
*default compress
src-all
ports-all tag=.
```

daha sonrada cvsup kurulumu:

```
[root@alchemist ~]# cd /usr/ports/net/cvsup-without-gui
[root@alchemist ~]# make install
```

herşey yonunda giderse cvsup komutunu çalıştırarak sisteminizi en güncel hale getirebilirsiniz.

```
[root@alchemist ~]# cvsup /etc/stable-supfile
```

Çekirdek Optimizasyonu

Artık kaynak kodlarımızı en güncel haline getirdiğimize göre çekirdek derleme işine girebiliriz. FreeBSD çekirdeği /usr/src/sys altındadır. Buradan sistem mimarinize göre bir seçim yapmanız gerekiyor. Ben 64 bitlik AMD kullandığım için çekirdek yapılandırma dosyasının yeri /usr/src/sys/amd64/conf altında. I386 mimarisi için amd64 yerine i386 dizinine girmeniz gerekiyor. Çekirdek derlemenin nasıl yapılacağı FreeBSD el kitabında mevcut.

#Çoklu islemci desteği 2 veya daha fazla işlemciniz varsa bu satırı ekleyin. Hyperthreading destekli işlemciler içinde bu satırı eklemek gerekiyor:

```
options SMP
```

nmap OS algılamasından korunmak için TCP'de SYN ve FIN set edilmişse paketi drop eder:

```
options TCP_DROP_SYNFIN
```

#ethernet kesme (interrupt) işini işletim sistemi kontrolüne bırakın.Daha fazla bilgi: man polling

```
Options DEVICE_POLLING
```

Device polling sıklığı 100Mbit ağ arabirimi için 1000 veya 2000 değerini kullanabilirsiniz.

```
options HZ=1000
```

#MAXDSIZ sistemde tek bir programın kullanabileceği maksimum bellek miktarıdır. Normalde bu limit 512 MByte'tır. Bind yazılımının kullanacağı bellek miktarı 512 MByte`tan fazla olabileceği için bu limit 3072 Mbyte'a yükseltildi:

```
options MAXDSIZ=(3072UL*1024*1024)
options MAXSSIZ=(1024UL*1024*1024)
options DFLDSIZ=(1024UL*1024*1024)
```

Eğer çok sayıda ağ bağlantısı açan bind, squid benzeri uygulamalarınız varsa mbuf alanını optimal kullanmak için bu değeri arttırmanız gerekiyor. Mbuf ağ arabirimden gelen ve gönderilen data paketlerinin geçici depolanması için kullanılır. Mbuf kullanımınızı netstat -m komutu ile görebilirsiniz:

```
options NMBCLUSTERS=32768
```

Son olarak sistemde bulunmayan donanımların desteğini çekirdekten çıkarmayı unutmayın.

/etc/rc.conf dosyasının ayarları

rc.conf dosyası FreeBSD'de açılış parametrelerini yazabileceğiniz dosyalardan biridir. Bu kısımda güvenlik ve optimizasyon için yapabileceğiniz ayarlar yer almaktadır:

#Icmp redirect mesajları drop etmek için:

```
icmp_drop_redirect="YES"
icmp_log_redirect="YES"
```

#Sisteminizin saatini otomatik olarak ntp sunucusundan ayarlayın:

```
ntpdate_program="/usr/sbin/ntpdate"
ntpdate_flags="ntp.ulak.net.tr"
```

#Çekirdek derlerken desteğini verdiğimiz TCP_DROP_SYNFIN özelliğini aktive edin:

```
tcp_drop_synfin="YES"
```

#Mail sunucunuzu dışarıdan mail almak için değil sadece içiriden dışarıya mail atmak için kullanacaksanız:

```
sendmail_enable="NO"
sendmail_submit_enable="YES"
```

/etc/sysctl.conf dosyası ayarları

FreeBSD'de çalışırken sistem parametrelerini sysctl komutu ile değiştirebilirsiniz. Yaptığınız ayarların kalıcı olması için bu ayarları sysctl.conf dosyası altına yazmanız gerekiyor:

Ağ kartı destekliyorsa DEVICE_POOLING opsiyonunu acar:

```
kern.polling.enable=1
```

Normal değeri 30 saniye servis dışı bırakma: DOS (“Denial of Service”) saldırılarından korumak için eğer 7500 msaniye icinde ACK alamazsa bağlantıyı kapar:

```
net.inet.tcp.msl=7500
```

max dosya dosya betimleyicisi (“file descriptor”) sayısı: (Daha fazla bilgi için FreeBSD handbook)

```
kern.maxfiles=65536
```

#Her bir işin (process) açabileceği maksimum dosya sayısı:

```
 kern.maxfilesperproc=32768
```

tcp gönderim pencere (“window”) aralığının max büyüklüğü normalde 32768, bu değeri ftp sunucusu benzeri büyük dosya transferi yapacak sunucularda 5242880 olarak değiştirebilirsiniz:

```
net.inet.tcp.sendspace=65536
net.inet.tcp.recvspace=65536
```

Yukarıdaki parametrenin udp için olanı:

```
net.inet.udp.recvspace=65535
```

#Max udp datagram büyüklüğü:

```
net.inet.udp.maxdgram=65535
```

```
kern.ipc.maxsockbuf=2097152
```

Gönderilecek icmp RST ve ICMP ulaşılamaz (“unreachable”) mesajlarının sayısı:

```
net.inet.icmp.icmplim=1300
```

#Yeni tcp bağlantıları kabul etmek için kullanılan dinleme kuyruğunun (“tcp listen queue”) maksimum sayısını belirler:

```
kern.ipc.somaxconn=4096
```

#Dizinlerin önbellekte nasıl tutulduğunu ayarlar. Çok G/Ç yapan sunucular 1'e set edildiğinde performanslarını arttırır:

```
vfs.vmiodirenable=1
```

Eğer kapalı olan bir tcp portuna bağlantı yapılmaya çalışılırsa kapalı olduğu bilgisini gonderme:

```
net.inet.tcp.blackhole=2
```

Yukarıdaki opsiyonun udp için olanı:

```
net.inet.udp.blackhole=1
```

#Bu opsiyon kapalı olduğunda IP paketleri sıralı olarak gönderildir. Opsiyon 1'e değiştirildiği zaman sıralama rasgele olur:

```
net.inet.ip.random_id=1
```

Sistem güvenlik açılarının takibi ve yamanması

FreeBSD’de sisteminizde açık olup olmadığını “portaudit” komutu ile takip edebilirsiniz. Çıkan açıkların yamanmasını portupgrade ile açık çıkan paketi bir üst sürümüne çıkararak yapabilirsiniz.

Portaudit kurulumu ve veri tabanının oluşturulması:

```
[root@alchemist ~]# cd /usr/ports/security/portaudit
[root@alchemist ~]# make install
[root@alchemist ~]# portaudit –Fda
```

Eğer sisteminizdeki hernangi bir açık yoksa:

```
[root@alchemist ~]# portaudit
0 problem(s) in your installed packages found.
```

Eğer açıklarınız varsa size paket ismini ve çıkan açığın açıklamasını belirtir:

```
[root@alchemist ~]# portaudit
Affected package: freetype2-2.1.10_3
Type of problem: freetype -- LWFN Files Buffer Overflow Vulnerability.
Reference: <http://www.FreeBSD.org/ports/portaudit/b975763f-5210-11db-8f1a-
000a48049292.html>
```

Portupgrade kurulumu:

```
[root@alchemist ~]# cd /usr/ports/sysutils/portupgrade
[root@alchemist ~]# make install
```

Portupgrade kurulduktan sonra açık olan freetype2-2.1.10_3 paketini yamamak için:

```
[root@alchemist ~]# portupgrade freetype2-2.1.10_3
```

Sisteminizde hangi portların dinlendiğini kontrol edilmesi.

Açık portların kontrolü için netstat ve sockstat komutlarını kullanabilirsiniz:

```
[root@alchemist ~]# netstat –an
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  10.10.10.2.50833   10.10.40.6.143       ESTABLISHED
tcp4       0      0  10.10.10.2.62151   10.10.40.6.143       ESTABLISHED
tcp4       0      0  10.10.10.2.63517   10.10.40.6.22        ESTABLISHED
tcp4       0      0  10.10.10.2.50090   10.10.40.251.22      ESTABLISHED
tcp4       0      0  10.10.10.2.51392   10.10.40.32.22       ESTABLISHED
tcp4       0      0  10.10.10.2.62552   192.168.3.1.22      ESTABLISHED
tcp4       0      0  10.10.10.2.54043   192.168.3.1.22      ESTABLISHED
tcp4       0      0  10.10.10.2.56185   192.168.3.1.22      ESTABLISHED
tcp4       0      0  10.10.10.2.50897   10.10.40.251.22      ESTABLISHED
tcp4       0      0  10.10.10.2.57970   10.10.40.251.22      ESTABLISHED
tcp4       0      0  10.10.10.2.56438   10.10.40.23.22       ESTABLISHED
tcp4       0      0  10.10.10.2.65147   10.10.40.23.22       ESTABLISHED
tcp4       0      0  10.10.10.2.49602   10.10.40.97.3389     ESTABLISHED
tcp4       0      0  127.0.0.1.37435        *.*                  LISTEN
tcp4       0      0  *.21000                *.*                  LISTEN
tcp4       0      0  *.111                  *.*                  LISTEN
tcp6       0      0  *.111                  *.*                  LISTEN
udp4       0      0  10.10.10.2.52014   10.10.40.32.514
udp4       0      0  *.514                  *.*
udp6       0      0  *.*                    *.*
udp4       0      0  *.609                  *.*
udp4       0      0  *.111                  *.*
udp6       0      0  *.1023                 *.*
udp6       0      0  *.111                  *.*
```

Bu tablodan çıkarılacak bilgi 21000,514 ve 609 numaralı portların Ipv4, 111 ve 1023 IPv6 olarak açık olduğudur. Bu portları kullanan programları öğrenmek için sockstat komutunu kullanabilirsiniz. IPv4 soketlerini görmek için:

```
[root@alchemist ~]# sockstat -4
USER     COMMAND    PID   FD PROTO  LOCAL ADDRESS         FOREIGN ADDRESS
root     ssh        78447 3  tcp4   10.10.10.2:63517  10.10.40.6:22
root     ssh        59911 3  tcp4   10.10.10.2:50090  10.10.40.251:22
root     ssh        88959 3  tcp4   10.10.10.2:51392  10.10.40.32:22
root     ssh        72973 3  tcp4   10.10.10.2:62552  192.168.3.10:22
root     ssh        72854 3  tcp4   10.10.10.2:54043  192.168.3.10:22
root     ssh        72810 3  tcp4   10.10.10.2:56185  192.168.3.10:22
root     ssh        65394 3  tcp4   10.10.10.2:50897  10.10.40.251:22
root     ssh        63695 3  tcp4   10.10.10.2:57970  10.10.40.251:22
root     thunderbir 60054 27 tcp4   10.10.10.2:50833  10.10.40.6:143
root     thunderbir 60054 30 tcp4   10.10.10.2:62151  10.10.40.6:143
root     ssh        83966 3  tcp4   10.10.10.2:56438  10.10.40.23:22
root     ssh        83942 3  tcp4   10.10.10.2:65147  10.10.40.23:22
```

```
root     rdesktop   4019  4  tcp4   10.10.10.2:49602  10.10.40.97:3389
root     kdeinit    647   10 tcp4   127.0.0.1:37435       *:*
root     syslog-ng  538   6  udp4   *:514                 *:*
root     syslog-ng  538   8  udp4   10.10.10.2:52014  10.10.40.32:514
root     sshd       449   3  tcp4   *:21000               *:*
root     rpcbind    386   9  udp4   *:111                 *:*
root     rpcbind    386   10 udp4   *:609                 *:*
root     rpcbind    386   11 tcp4   *:111                 *:*
```

514'üncü portu syslog-ng, 21000`i sshd, 111, 609 portunda rpcbind tarafından dinlenildiğini böylece görmüş oluyoruz.

Aynı komutla IPv6 dinleyen soketleri görmek için: (IPv6 hakkında daha fazla bilgi için www.ipv6.org.tr)

```
[root@alchemist ~]# sockstat -6
USER     COMMAND    PID   FD PROTO  LOCAL ADDRESS         FOREIGN ADDRESS
root     rpcbind    386   4  udp6   *:*                   *:*
root     rpcbind    386   6  udp6   *:111                 *:*
root     rpcbind    386   7  udp6   *:1023                *:*
root     rpcbind    386   8  tcp6   *:111                 *:*
```

Bu durumda örneğin nfs ile ilgili bir işleminiz yoksa rpcbind'ı kapatmanız mantıklı olacaktır.

Sisteminizde kritik dosyaların değişip değişmediğinin kontrolü:

Sisteminiz kırıldığı zaman saldırganın yapacağı işlemlerden ilki daha sonra bağlanmak için kendisine bir açık kapı (backdoor) bırakmak olacaktır. Bu tür bir işlemi yapmanın en kısa yolu sshd programını yamamak olabilir. Bunun yanında sistem log dosyalarından kendini otomatik olarak silen ve görünmez kılan rootkit yazılımları da sisteminize yüklenebilir. Bu durumu sistem (şanslıysanız ☺) dosyalarında yapılan değişikliklerden fark edebilirsiniz.

Dosyaların değişip değişmediğinin kontrolünü açık kaynak kodlu programlarla yapabileceğiniz gibi (örneğin tripwire, aide), kendi yazdığınız betiklerle de yapabilirsiniz. Sisteme saldıran kişi bu tür programların kurulduğunu farkedip devre dışı bırakabileceği için kanımca en iyi yöntem kendinizin yazdığı standart olmayan bir veri doğrulama (checksum) programı kullanmanız.

Aşağıda yazdığım betik DIRLIST değişkeni altında tanımladığınız dizinlerin md5 toplamlarını alarak değişiklik olduğu zaman ADMIN değişkeni altında tanımladığınız kişiye mail atarak durumu bildirir. Checksum dosyalarının saklanması için ben /usr/local/man11 dizini tercih ettim. Bu dizin saldırganın dikkatini çekmeyecek herhangi bir yer olabilir. Find, diff, md5 komutların yerlerini sisteminize göre değiştirin.

```
#!/bin/bash
#onur at ulakbim.gov.tr (18/7/2005)
#Degiskenler.
FIND="/usr/bin/find"
MD5="/sbin/md5"
DIFF="/usr/bin/diff"
DIRLIST="/etc/ /usr/local/bin/  /bin /sbin /usr/local/etc /usr/local/sbin"
ILKDB="/usr/local/man/man11/ydb"
SONDB="/usr/local/man/man11/edb"
TEMPF1="/tmp/aaa.$$"
TEMPF2="/tmp/bbb.$$"
ADMIN="ornek@ulakbim.gov.tr"
```

```
MAILY="`uname -ns`  sistemi uzerinde degisen dosyalar\n"

## Bu kisimdan sonra degisiklik yapmayin..

if [ ! -f "$ILKDB" ]
        then
        echo "program ilk defa calistiriliyor"
                for i in $DIRLIST
                        do
                                $FIND $i -exec $MD5 {} \; >> $ILKDB 2> /dev/null
                        done
        else
                for i in $DIRLIST
                        do
                                $FIND $i -exec $MD5 {} \; >> $TEMPF1 2> /dev/null
                        done

        $DIFF $ILKDB $TEMPF1  --side-by-side --suppress-common-lines  >     $TEMPF2
        if [ -s $TEMPF2 ]
                then
                echo -e " $MAILY `cat $TEMPF2`" | mail $ADMIN
        fi
        rm -f $TEMPF1
        rm -f $TEMPF2

fi
```

Bu betiğinin ismini dikkat çekmeyecek birşey koyup (namedstat) cron'a 6 saatte bir çalışacak şekilde ekleyin:

```
0 0,6,12,18 * * * /usr/sbin/namedstat
```

Chkrootkit programı ile rootkit taranması

Rootkitler, veri doğrulama ("checksum") programlarını işlevsiz hale getirebiliyorlar. İlave bir önlem olarak sisteminizde rootkit olup olmadığının kontrolü için chkrootkit programınıda kullanabilirsiniz. Kurulumu:

```
[root@alchemist ~]# cd /usr/ports/security/chkrootkit
[root@alchemist ~]# make install
```

Programı çalıştırdığınızda rootkit taraması yapılıp raporu ekrana basılıyor.

Sistem loglarının ortak makinede toplanması, syslog-ng kurulumu

Sunucunuz kırıldığı zaman saldırganın yapacağı ilk şeylerden birinin sisteminizdeki loglardan kendisine ait kayıtları temizlemek olduğunu daha önceden söyledik. Tüm sunucularınızın loglarını sunucu üzerinde değilde başka bir log sunucusunda tutmak iyi bir çözüm olabilir. Bunun için Unix'ler ile standart olarak gelen syslogd yerine syslog-ng programını kullanabilirsiniz.

Syslog-ng kurulumu için kullandığınız unix dağıtımı ile gelen paket yönetim sistemini kullanabileceğiniz gibi kaynak kodundan kendinizde derleyebilirsiniz. Kaynak kodunun nasıl derleneceği, dökümanın yazıldığı esnada en son kararlı sürümü 1.6 olduğu için örnek kurulum bu versiyona göre yapıldı. Syslog-ng kurabilmeniz için libol ve libnet kütüphanelerine ihtiyaç duyacağız, gerekli dosyalar:

```
http://www.balabit.com/products/syslog_ng/
```

adresi altında var, kurulum için aşağıdaki dosyalar indirildi:

```
http://www.balabit.com/downloads/libol/0.3/libol-0.3.18.tar.gz
http://www.packetfactory.net/libnet/dist/libnet.tar.gz
http://www.balabit.com/downloads/syslog-ng/1.6/src/syslog-ng-1.6.11.tar.gz
```

Kurulum için:

```
bash-3.1# tar –xzvf libnet.tar.gz
bash-3.1# cd libnet
bash-3.1# ./configure
bash-3.1# make
bash-3.1# make install
bash-3.1# cd ..
bash-3.1# tar -xzvf libol-0.3.18.tar.gz
bash-3.1# cd libol-0.3.18
bash-3.1# ./configure
bash-3.1# make
bash-3.1# make install
bash-3.1# cd ..
bash-3.1# tar -xzvf syslog-ng-1.6.11.tar.gz
bash-3.1# ./configure --prefix=/ /usr/local/syslog-ng-1.6.11
bash-3.1# make
bash-3.1# make install
bash-3.1# ln –s /usr/local/syslog-ng-1.6.11 /usr/local/syslog-ng
```

Bu kadar uğraşmak yerine portlar altından kurulumda yapabilirsiniz:

```
[root@alchemist ~]# cd  /usr/ports/sysutils/syslog-ng
[root@alchemist ~]# make install
```

Syslog-ng.conf dosyasının ayarları

Syslog-ng istemci (client) orarak ayarları: /etc/hosts dosyası altında syslog-ng sunucunuzu loghost olarak tanımlayan aşağıdaki satırları ekleyin. Log sunucunuzun logcu.ulakbim.gov.tr (10.10.10.2) ise örneğin:

```
10.10.10.2          logcu logcu.ulakbim.gov.tr  loghost
```

eklemeniz gerekli. Syslog-ng.conf formatında source, destination, filter, log benzeri ayar kısımları var. Bizim ilgilendiğimiz source, destination ve log.

Source kısmı syslog-ng.conf dosyasının ilk kısmında bulunur ve sistem loglarının nereden okunacağını gösterir. Örnek syslog.conf dosyasını kopyaladığınızda bu kısım zaten olacağı için herhangi bir değişiklik yapılmasına gerek yok. Sadece source ismi olarak ne verildiği önemli, linux örnek dosyasına bakarsak:

```
source s_all {
        # message generated by Syslog-NG
        internal();
        # standard Linux log source (this is the default place for the syslog()
        # function to send logs to)
        unix-stream("/dev/log");
        # messages from the kernel
        file("/proc/kmsg" log_prefix("kernel: "));
        # use the above line if you want to receive remote UDP logging messages
        # (this is equivalent to the "-r" syslogd flag)
```

```
        # udp();
};
```

Burada tek kullanacağımız source ismi olan “s_all”

syslog-ng’nin logları loghost sunucusuna basması için syslog.conf dosyasının içinde destination olarak loghost’u tanımlamanız gerekiyor. Örneğin d_remote adında bir destination yaratmak için:

```
destination d_remote { udp("loghost" port(514)); };
```

yazmanız yeterli. Burada d_remote destination tanımının ismi, diğer kısımda logların loghost olarak /etc/hosts dosyasında tanımlanan sunucunun 514 numaralı portuna gönderileceğini belirtiyor. Log işlemin nereye yapılacağı ile ilgili olarak log tanımını eklemek için:

```
log { source(s_src); destination(d_remote); };
```

yazmanız yeterli.

Syslog-ng sunucu (server) orarak ayarları: Syslog-ng programını sunucu olarak çalıştırmak için aşağıdaki örnek ayar dosyasından source kısmını kullandığınız unix türevine göre değiştirmeniz gerekiyor. Bunun için ; ile ayrılmış olan source satırların sonuna:

```
udp(ip(0.0.0.0) port(514));
```

tanımını eklemeniz gerekli. İşletim sistemlerine göre örnek source tanımları:

Solaris:

```
source s_sys { sun-streams ("/dev/log"); internal(); udp(ip(0.0.0.0) port(514)); };
```

FreeBSD:

```
source src { unix-dgram("/var/run/log");
             unix-dgram("/var/run/logpriv" perm(0600));
              internal(); file("/dev/klog"); udp(ip(0.0.0.0) port(514)); };
```

Linux:

```
source s_all {
        # message generated by Syslog-NG
        internal();
        # standard Linux log source (this is the default place for the syslog()
        # function to send logs to)
        unix-stream("/dev/log");
        # messages from the kernel
        file("/proc/kmsg" log_prefix("kernel: "));
        # use the above line if you want to receive remote UDP logging messages
        # (this is equivalent to the "-r" syslogd flag)
         udp(ip (0.0.0.0) port(514));
};
```

Log dosyalarının nereye yazılacağını destination kısmında belirtiyoruz. Aşağıdaki örnek dosyadaki;

```
destination d_1     { file("/loglar/$HOST/$FACILITY/log.$YEAR$MONTH$DAY");};
```

satırı dosyaların /loglar dizini altına yazılacağını gösteriyor. Bu satırdaki dizini sistemdeki logları koyacağınız dizine göre değiştirmeniz gerekiyor.

Log sunucusu için örnek syslog-ng.conf dosyası:

```
 options { sync (0);
          time_reopen (10);
          log_fifo_size (1000);
          long_hostnames (off);
          use_dns (no);
          use_fqdn (no);
          create_dirs (yes);
          keep_hostname (yes);
        };

source "işletim sistemine göre yukarıadaki örnekten kopyala yapıştır yapılacak"
destination d_1     { file("/loglar/$HOST/$FACILITY/log.$YEAR$MONTH$DAY");};
filter f_1 { not match('dropped'); };
# Source ismi olarak yukarda ne kullandıysanız bu kısımdada source(????) kısmına onu
# yazmanız gerekiyor
log { source(????); filter(f_1); destination(d_1); };
```

Syslog-ng programının çalıştırılması

Syslog-ng programını çalıştırmadan önce syslogd programını öldürmeniz ve artık syslog-ng kullanılacağı için syslogd programını açılıştan kaldırmak gerekiyor. Syslog-ng programını çalıştırmak için syslog.conf dosyasını düzenledikten sonra /etc altına kopyalayın ve komut satırından:

```
bash-3.1# /usr/local/syslog-ng/sbin/syslog-ng -c /etc/syslog-ng.conf pidfile=/tmp/syslog-
ng.pid
```

yazarak çalıştırın. Bu satırın işletim sisteminin her açılışında çalışacak şekilde unix sisteminizin boot betiklerine eklenmesi gerekiyor. FreeBSD`de /etc/rc.conf dosyasının altında aşağıdaki satırları eklemeniz yeterli:

```
syslog_ng_enable="YES"
syslogd_enable="NO"
```

Nelerden Bahsetmedik.

Güvenlik duvarı kurulumundan bahsetmedik. Sisteminize IPFILTER ile güvenlik duvarı kurup yapılandırabilirsiniz.Güvenlik duvarı kurlumu ile ilgili bolca türkçe döküman olduğu için bu konuya girmedim.

Bunun yanında güvenlik açıklarının takibi için güvenlik listelerine üye olmakta fayda var. Geniş bir liste için Ulak-CSIRT web sayfasına göz atmanızı öneririm (http://csirt.ulakbim.gov.tr/link.uhtml)

Bu listenin içinden securiteam (www.securityfocus.com) ve CERT (www.cert.org) en azından üye olmanız gereken iki liste.