

## FreeBSD üzerinde SNMP ve RRDTool kullanımı

Bu belgede özet olarak FreeBSD üzerinde SNMP ve RRDTool kullanarak sunucu istatistiklerini görsel olarak görebilmemiz için gereken alt yapının oluşturulması anlatılacaktır.

### Gerekli Yazılımların Kurulması

İstatistikleri oluşturmak için FreeBSD portlarında bulunan **net-snmp** ve **rrdtool** paketlerinin kurulması gerekmektedir.

```
# cd /usr/ports/net-mgmt/net-snmp && make install clean
# cd /usr/ports/net/rrdtool && make install clean
```

Bu komutlar aracılığı ile net-snmp ve rrdtool paketleri sisteme kurulduktan sonra snmp sunucunun ayarlarının yapılması gerekmektedir.

FreeBSD üzerinde SNMP sunucunun sistem açıldığında otomatik olarak başlaması için /etc/rc.conf dosyasına

```
snmp_enable="YES"
```

satırının eklenmesi gerekmektedir.

İlk olarak rc.conf dosyasına bu satır eklendikten sonra SNMP sunucunun ayarlarının yapılması için **snmpconf** komutu kullanılmaktadır.

```
devel~# snmpconf -i

I can create the following types of configuration files for you.
Select the file type you wish to create:
(you can create more than one as you run this program)

  1:  snmpd.conf
  2:  snmptrapd.conf
  3:  snmp.conf

Other options: quit

Select File:
```

Bizim için gerekli olan snmpd olduğu için snmpd.conf(5) u seçmemiz gerekmektedir. Snmpd.conf seçeneği seçildikten sonra aşağıdaki menü gelecektir.

The configuration information which can be put into snmpd.conf is divided into sections. Select a configuration section for snmpd.conf that you wish to create:

- 1: Access Control Setup
- 2: Extending the Agent
- 3: Monitor Various Aspects of the Running Host
- 4: Agent Operating Mode
- 5: System Information Setup
- 6: Trap Destinations

Other options: finished

Bu menüde 1inci seçenek Access Control Setup seçilmelidir.

Section: Access Control Setup

Description:

This section defines who is allowed to talk to your running snmp agent.

Select from:

- 1: a SNMPv3 read-write user
- 2: a SNMPv3 read-only user
- 3: a SNMPv1/SNMPv2c read-only access community name
- 4: a SNMPv1/SNMPv2c read-write access community name

Other options: finished, list

Select section:

Bu menüde istemci tarafında kullanılacak SNMP istemci için kullanılması gereken SNMP versiyon bilgisi girilecektir.

SNMPv1 de topluluk(community) ismi ile işlemler yapılmaktadır. SNMP v2 ile kullanıcı yetkilendirme, SNMP v3 ile de şifreleme gibi özellikler gelmektedir.

Bizim için bu menüde en uygun olan 3ncü seçenek olan “a SNMPv1/SNMPv2c read-only access community name” seçeneğidir. Bu seçenek ile hem SNMP v1 hem de SNMP v2c desteklenecektir. Ek olarak SNMP isteklerinin sadece okuma modunda ve topluluk adı ile yapılması sağlanacaktır.

Configuring: rocommunity

Description:

a SNMPv1/SNMPv2c read-only access community name  
arguments: community [default|hostname|network/bits] [oid]

The community name to add read-only access for:

Burada “**rocommunity**” olarak Read-Only Community ismi sorulmaktadır. Topluluk adı olarak seçtiğimiz ismin, istemci tarafından da girilmesi gerekmektedir. Örneğin isim olarak “**enderunix**” seçelim.

```
The community name to add read-only access for: enderunix
The hostname or network address to accept this community name from
[RETURN for all]:
```

Burada topluluk adı olarak “**enderunix**” girildikten sonra erişim için bilgisayar adı veya ağ adresi istenmektedir. Bu kısım şimdilik konumuzun dışında olduğu için boş geçilecektir.

```
The OID that this community should be restricted to [RETURN for no-
restriction]:
```

Daha sonra istemci tarafından yapılacak tüm isteklerin kısıtlanıp kısıtlanmaması için yukarıdaki soru gelecektir. Bu soruya uygun MIB aileleri yazarak sadece o MIB ailesi altındaki MIB değerlerinin kullanılması sağlanabilir. Bu kısımda boş geçilecektir.

Daha sonra gelen her 2 menüye de “**finished**” konfigürasyon dosyalarının seçildiği ekrana gelinir. Bu menüde “**quit**” yazılarak çıkılır.

Finished Output: rocommunity **enderunix**

Section: Access Control Setup

Description:

This section defines who is allowed to talk to your running snmp agent.

Select from:

- 1: a SNMPv3 read-write user
- 2: a SNMPv3 read-only user
- 3: a SNMPv1/SNMPv2c read-only access community name
- 4: a SNMPv1/SNMPv2c read-write access community name

Other options: finished, list

Select section: **finished**

The configuration information which can be put into snmpd.conf is divided into sections. Select a configuration section for snmpd.conf that you wish to create:

- 1: Access Control Setup
- 2: Extending the Agent
- 3: Monitor Various Aspects of the Running Host
- 4: Agent Operating Mode
- 5: System Information Setup
- 6: Trap Destinations

Other options: finished

Select section: **finished**

I can create the following types of configuration files for you. Select the file type you wish to create:

(you can create more than one as you run this program)

- 1: snmpd.conf
- 2: snmptrapd.conf
- 3: snmp.conf

Other options: quit

Select File: **quit**

The following files were created:

snmpd.conf installed in /usr/local/share/snmp

Bu işlemden sonra snmp.conf dosyası /usr/local/share/snmp dizini altında oluşturulmuştur.

```
devel~# cat /usr/local/share/snmp/snmpd.conf
#####
##
#
# snmpd.conf
#
# - created by the snmpconf configuration program
#
#####
##
# SECTION: Access Control Setup
#
# This section defines who is allowed to talk to your running
# snmp agent.

# rocommunity: a SNMPv1/SNMPv2c read-only access community name
# arguments: community [default|hostname|network/bits] [oid]

rocommunity enderunix

devel~# /usr/local/etc/rc.d/snmpd.sh start
Starting snmpd.
devel~#
```

Son verilen komut ile de SNMP sunucunun başlatılması sağlanır.

## RRDTool Kullanımı

```
# mkdir /usr/local/rrd && cd /usr/local/rrd
# rrdtool create bandwidth.rrd -s 300 --start N DS:in:COUNTER:600:U:U \
DS:out:COUNTER:600:U:U RRA:AVERAGE:0.5:1:432
```

komutları ile /usr/local/rrd dizini oluşturulur ve trafik ölçümü için gerekli olan rrd (Round-Robin Database) dosyası oluşturulur.

1. **rrdtool create bandwidth.rrd** : komutu yeni bir RRD dosyası oluşturmak için kullanılır.
2. **-s 300** : Veritabanındaki kayıtları arasındaki süreyi gösterir.
3. **--start N** : Bu parametre ile rrd dosyasının başlangıç zamanı verilir. N, now anlamına gelmektedir.
4. **DS:in:COUNTER:300:U:U** : Bu seçenek ile rrd dosyasında tutulacak veritip tanımlanır.
  - o **DS** : Veri kaynağının başladığını gösterir.
  - o **in** : Veri kaynağının adıdır.
  - o **COUNTER** : Bazı veritipleri sadece artan tipte, azalan tipte yada değişken tipte olabilir. COUNTER kullanımı sadece artan tipte olduğunu gösterir.
  - o **600** : Veri girilecek sıklığı gösterir.
  - o **U:U** : Opsiyonel olarak en düşük ve en büyük değerler girilebilir.
5. **RRA:AVERAGE:0.5:1:432** : Create komutunun son parçası olarak Round-Robin Archive kullanılabilir. Veritabanında ek bir değer tutulması için kullanılır.
  - o **RRA** : Round-Robin Archive değerini tanımlar

- **AVERAGE** : RRA in bir parçası olarak bu alanın nasıl hesaplanacağını gösterir.
  - **0.5** : xff veya xfiles faktörü olarak adlandırılır.
  - **1** : RRA değerinin kaç kayıta bir hesaplanacağını gösterir.
  - **432** : Veritabanında kaç kaydın tutulacağını gösterir.
6. Birden fazla RRA değeri tanımlanabilir.

rrdtool dump komutu ile veritabanının içeriği görülebilir.

```
devel~# rrdtool dump bandwidth.rrd | more
<!-- Round Robin Database Dump --><rrd> <version> 0003 </version>
  <step> 300 </step> <!-- Seconds -->
  <lastupdate> 1159475292 </lastupdate> <!-- 2006-09-28 23:28:12 EEST -->

  <ds>
    <name> in </name>
    <type> COUNTER </type>
    <minimal_heartbeat> 300 </minimal_heartbeat>
  ...
  <database>
    <!-- 2006-09-27 11:35:00 EEST / 1159346100 --> <row><v> NaN </v><v> NaN </v></row>
    <!-- 2006-09-27 11:40:00 EEST / 1159346400 --> <row><v> NaN </v><v> NaN </v></row>
    <!-- 2006-09-27 11:45:00 EEST / 1159346700 --> <row><v> NaN </v><v> NaN </v></row>
```

## Round Robin veritabanına kayıt eklenmesi

RRD dosyamıza yeni bir kayıt eklemek için rrdupdate komutunu kullanıyoruz. rrdupdate komuna parametre olarak rrd dosyamızı ve rrd dosyamızda bulunan alanların değerlerini vermemiz gerekiyor.

```
# rrdupdate bandwidth.rrd timestamp:değer1:değer2
```

Şeklinde kullanılabilir. Burada timestamp değerini N olarak verip, şu anı kastediyor olacağız.

Burada değer1 ve değer2 alanlarına uzak sunucumuzda yada localhost tan SNMP istemci ile aldığımız değerleri vermemiz gerekiyor.

```
# devel~# snmpget -v 1 -c enderunix -Oqv localhost IF-
MIB::ifInOctets.1 IF-MIB::ifOutOctets.1
3157342182
3755828453
```

komuta ile 1nci arayüz için gerekli değerleri alabiliriz. Burada izlenilmek istenen arayüze göre gerekli değişiklik yapılmalıdır.

```
devel~# snmpwalk -v1 -c enderunix -Os localhost IF-MIB::ifDescr
ifDescr.1 = STRING: fxp0
ifDescr.2 = STRING: lp0
ifDescr.3 = STRING: lo0
ifDescr.4 = STRING: ppp0
ifDescr.5 = STRING: sl0
```

Komutu ile tüm arayüzler görülebilir. Buradan seçilecek uygun arayüz snmpget ile kullanılabilir.

```
# devel~# snmpget -v 1 -c enderunix -Oqv localhost IF-MIB::ifInOctets.1
IF-MIB::ifOutOctets.1
3157342182
3755828453
```

snmpget ile alınan bu değerleri rrdupdate e vermemiz gerekiyor.

```
# rrdupdate /usr/local/rrd/bandwidth.rrd N:\
  `/usr/local/bin/snmpget -v 1 -c enderunix -Oqv localhost IF-
MIB::ifInOctets.1`\
  `/usr/local/bin/snmpget -v 1 -c enderunix -Oqv localhost IF-
MIB::ifOutOctets.1`
```

komutu ile 1nci arayüz için hem in değerini hem de out değerlerini girmiş oluyoruz.

Üstteki komut, crontab aracılığı ile 5 dakikada bir çalışacak şekilde ayarlanırsa, rrd dosyamız sürekli güncellenecektir.

```
devel:# rrdupdate /usr/local/rrd/bandwidth.rrd \
N:`/usr/local/bin/snmpget -v 1 -c enderunix -Oqv localhost \
IF-MIB::ifInOctets.1`:`/usr/local/bin/snmpget -v 1 -c enderunix \
-Oqv localhost IF-MIB::ifOutOctets.1`
```

komutunu kullanarak rrd dosyamıza ilk kaydımızı ekleyelim.

```
devel/home/siseci/snmp# rrdtool dump bandwidth.rrd | tail
<!-- 2006-10-12 21:30:00 EEST / 1160677800 --> <row><v> NaN </v><v> NaN </v></row>
<!-- 2006-10-12 21:35:00 EEST / 1160678100 --> <row><v> NaN </v><v> NaN </v></row>
<!-- 2006-10-12 21:40:00 EEST / 1160678400 --> <row><v> NaN </v><v> NaN </v></row>
<!-- 2006-10-12 21:45:00 EEST / 1160678700 --> <row><v> NaN </v><v> NaN </v></row>
<!-- 2006-10-12 21:50:00 EEST / 1160679000 --> <row><v> NaN </v><v> NaN </v></row>
<!-- 2006-10-12 21:55:00 EEST / 1160679300 --> <row><v> NaN </v><v> NaN </v></row>
<!-- 2006-10-12 22:00:00 EEST / 1160679600 --> <row><v> 3.1277885500e+02 </v><v>
6.7389431013e+02 </v></row>
</database>
</rra>
</rrd>
```

Yukarıda görüldüğü gibi **rrdtool dump** komutunu kullanarak round-robin veritabanımıza eklenen kayıtları görebiliriz.

## RRD Dosyasından Grafik Oluşturmak

Gerekli programları sistemimize yükleyip, gerekli ayarları yaptık. Artık veritabanımıza bilgi de girebiliyoruz. RRDtool yardımı ile MRTG benzeri grafikler oluşturabiliriz. RRDtool ile PNG, GIF ve GD formatlarında grafikler oluşturabiliriz. Ancak lisans özellikleri ve taşınabilirliğinden dolayı PNG yi seçmek en iyi çözüm olacaktır.

**rrdtool graph** komutu ile rrd dosyamızdan istediğimiz grafikleri oluşturabiliriz. Bu komutun parametreleri biraz fazla olduğu için bunun bir betiğin için yazıp kullanmak bize kolaylık sağlayacaktır.

```
/usr/local/bin/rrdtool graph /usr/local/rrd/bandwidth.png -a PNG \  
-h 125 \  
-s -129600 \  
-v "Data Throughput" \  
'DEF:in=/usr/local/rrd/bandwidth.rrd:in:AVERAGE' \  
'DEF:out=/usr/local/rrd/bandwidth.rrd:out:AVERAGE' \  
'CDEF:kbin=in,1024,/' \  
'CDEF:kbout=out,1024,/' \  
'AREA:in#00FF00:Bandwidth In' \  
'LINE1:out#0000FF:Bandwidth Out\j' \  
'GPRINT:kbin:LAST:Last Bandwidth In\: %3.2lf KBps' \  
'GPRINT:kbout:LAST:Last Bandwidth Out\: %3.2lf KBps\j' \  
'GPRINT:kbin:AVERAGE:Average Bandwidth In\: %3.2lf KBps' \  
'GPRINT:kbout:AVERAGE:Average Bandwidth Out\:%3.2lf KBps\j'
```

Bu komutu açıklayalım:

- DEF: RRD dosyamızdan almak istediğimiz alanları belirtmek için kullanılır.
- CDEF: RRDTool, grafik oluşturulmadan önce gelen veriler üzerinde matematiksel işlemler yapmamıza izin verir. Buradan kbin ve kbout değerlerini RRD dosyamızdaki in ve out değerlerinden oluşturuyoruz. Burada 1024 ve / ile alınan değerlerin 1024 e bölüneceğini yani byte cinsinden değilde kilobyte cinsine çevrilip kullanılacağı belirtilmiştir.
- AREA ve LINE: Tipik olarak AREA ya in, LINE a out değerleri yazılır. \j tipik olarak hizalanmış görüntü verir.
- GPRINT: Grafikteki lejandın altındaki ortalama değerlerini yazmak için kullanılır.

Yukarıdaki komutu crontab a eklediğimiz **rrdupdate** betiğinin altına ekleyerek 5 dakikada bir hem RRD dosyamızın güncellenmesini hemde grafiğimizin oluşturulmasını sağlayabiliriz.

ifInOctets.1 ve ifOutOctets.1 ile sistemdeki 1nci Ethernet kartının trafik bilgilerini aldık. Bunun dışında

- UCD-SNMP-MIB::laLoadInt.1, UCD-SNMP-MIB::laLoadInt.2 and UCD-SNMP-MIB::laLoadInt.3  
Sistem yükünün 1, 5 ve 15 dakikalık değerlerini alabiliriz. Bu değerleri 100 ile çarpmak bize tam sayılar verecektir.
- UCD-SNMP-MIB::memAvailReal.0  
Sistemdeki boş fiziksel bellek miktarını verir.



- **HOST-RESOURCES-MIB::hrSystemProcesses.0**  
Sistemde çalışan proses sayısını verir.

Bu değerler size yeterli olmuyorsa /usr/local/share/snmp/mibs dizinindeki dosyalara bakabilirsiniz.

Buradaki dosyalardan HOST-RESOURCES-MIB.txt dosyasından bir kaydı inceleyelim.

HOST-RESOURCES-MIB.txt:

hrSystemProcesses OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of process contexts currently loaded or  
running on this system."

::= { hrSystem 6 }

Buradaki hrSystemProcess değerini kullanmak için

**HOST-RESOURCES-MIB::hrSystemProcesses.0** değerini kullanmak gerekir. Burada en sona eklenen 0, bazı MIB değerlerinin birden fazla değer verebilmesinden dolayı eklenmiştir ve ilk değer anlamına gelir. Sistem yükünü almak için kullandığımız laLoadInt MIB'in deki gibi. Emin değilseniz 0 kullanarak deneyebilirsiniz.

## **Kaynaklar:**

- <http://www.freebsd.org>
- <http://oss.oetiker.ch/rrdtool/>
- <http://net-snmp.sourceforge.net/>
- <http://silverwraith.com/papers/freebsd-snmp.php>

## **BELGE HAKKINDA**

Bu belge "GNU Free Documentation Licence" ı ile kaynak gösterilmek ve önceden yazarından izin alınmak kaydıyla yeniden yayınlanabilir.

Belgedeki eksik, yanlış ya da geliştirilmesi gerektiğini düşündüğünüz yerleri lütfen yazarına e-posta ile bildiriniz.

Bu belgenin en güncel haline,  
<http://www.enderunix.org/>  
adresinden ulaşabilirsiniz.

Belgenin İlk Oluşturulma Tarihi: 28 Eylül 2006 Perşembe, 22:43:00  
Belgenin Son Güncellenme Tarihi: 12 Ekim 2006 Perşembe, 22:54:30

Tüm hakları Necati Ersen ŞİŞECİ'ye aittir.

## **YAZAR HAKKINDA**

**Necati Ersen ŞİŞECİ**  
EnderUnix Yazılım Geliştirme Takımı ~ Türkiye  
siseci <<<< at >>>> enderunix.org  
<http://www.enderunix.org/siseci>