



Vnstat - мониторинг сетевой трафик

Опубликовано muff в Вс, 2010-08-15 03:16

Продолжая эпопею считалок и мониторилок сетевых интерфейсов и трафика, рассмотрим утилиту **vnstat**. **Vnstat** - консольная утилита для учета сетевого трафика. Сохраняет историю сетевого трафика для выбранных интерфейсов. Трафик считается отдельно для каждого интерфейса (как входящий, так и исходящий). Доступна различная детализация отчетов: час, день, неделя, месяц...

Идем в порты и выполняем установку:

```
# cd /usr/ports/net/vnstat/ && make install clean && rehash
```

После установки утилита "любезно" предупредила, что необходимо внести содержимое файла **/usr/local/share/doc/vnstat/vnstat-cron** в **crontab**. Ну а если нас интересует информация о утилите, необходимо посетить ресурс <http://humdi.net/vnstat/> [1], либо же воспользоваться справочной системой FreeBSD.

Начинаем попытки "подружиться" с утилитой. Для начала скопируем пример конфигурационного файла непосредственно в рабочий конфигурационный файл:

```
# cp /usr/local/etc/vnstat.conf.sample /usr/local/etc/vnstat.conf
```

Следующим шагом редактируем **/usr/local/etc/vnstat.conf**, подстраивая его под свои нужды. Я же изменил только название интерфейса, на котором будем "слушать" трафик. Потом в конфигурационном файле находим опцию **DatabaseDir**, и согласно указанному значению, создаем базу данных (в моем частном случае это **/var/db/vnstat**, поскольку настройки по умолчанию этого параметра я не изменял).

```
# mkdir /var/db/vnstat
```

Добавляем запись в **cron** для автоматизации процесса сбора статистики и перезапускаем его:

```
# cat /usr/local/share/doc/vnstat/vnstat-cron >> /etc/crontab  
# killall -1 cron
```

Теперь проинициализируем интерфейсы, для которых будем собирать статистику:

```
# vnstat -u -i tun0  
Error: Unable to read database "/var/db/vnstat/tun0".  
Info: -> A new database has been created.  
# >vnstat -u -i vlan8  
Error: Unable to read database "/var/db/vnstat/vlan8".  
Info: -> A new database has been created.
```

Через некоторое время можно выбрать данные из коллектора:

```
# vnstat  
  
          rx      /      tx      /      total      /      estimated tun0:  
Aug '10      5 KiB /      4 KiB /      9 KiB /      0 KiB      today  
      5 KiB /      4 KiB /      9 KiB /      --  
  
vlan8:      Aug '10      992 KiB /      4.45 MiB /      5.42 MiB /      0 KiB  
      today      992 KiB /      4.45 MiB /      5.42 MiB /      --
```



Для просмотра загрузки интерфейса в реальном времени воспользуемся ключем **-l (--live)**:

```
# vnstat --live -i vlan8Monitoring vlan8... (press CTRL-C to stop)

rx:          0 kbit/s      3 p/s          tx:          4 kbit/s      3 p/s^C

vlan8 / traffic statistics

-----+----- rx | tx-----+-----
bytes          0.98 MiB | 4.29 MiB-----+-----
max          312 kbit/s
s | 1.86 Mbit/s average 42.58 kbit/s | 186.12 kbit/s
min          0 kbit/s | 0 kbit/s-----+-----
+-----+----- packets          3761 | 4361-----+-----
max          110 p/s
| 163 p/s average 19 p/s | 23 p/s m
in          0 p/s | 0 p/s-----+-----
time          3.15 minutes
```

Для того, чтобы узнать больше о возможностях **vnstat**, советую ознакомиться со страницами руководства пользователя. Я же наведу только некоторые примеры...

Чтобы вывести статистику за последних 24 часа (с почасовой детализацией), для интерфейса **vlan8**, команда будет выглядеть следующим образом:

```
# vnstat -h -i vlan8
```

Для вывода статистики с дневной детализацией, воспользуемся ключем **-d**:

```
# vnstat -d -i vlan8
```

Для вывода статистики за месяц, существует ключ **-m**:

```
# vnstat -m -i vlan8
```

А чтобы просмотреть статистику 10 самых загруженных дней, можно воспользоваться такой командой:

```
# vnstat --top10 -i vlan8
```

Скриншоты не предоставляю, поскольку в коллекторе еще слишком мало информации о собранном трафике (меньше суток).

Также, есть возможность воспользоваться **vnStat PHP Frontend** - написанный на PHP web-интерфейс для vnStat. Переходим в каталог **/usr/local/www** и скачиваем архив web-интерфейса:

```
# cd /usr/local/www/
# wget http://muff.kiev.ua/files/vnstat_php_frontend-1.5.1.tar.gz
```

Распаковываем скачанный архив:

```
# tar -xzf vnstat_php_frontend-1.5.1.tar.gz
```

Потом переименовываем каталог с vnStat PHP Frontend в более "дружелюбное" название:

```
# mv vnstat_php_frontend-1.5.1 vnstat
```



Правим конфигурационный файл web-интерфейса **/usr/local/www/vnstat/config.php** до следующего состояния:

```
# cat /usr/local/www/vnstat/config.php
```

```
<?php
error_reporting(E_ALL | E_NOTICE);
$locale = 'en_EN.UTF-8';
$language = 'en';
$iface_list = array('vlan8', 'tun0');
$iface_title['vlan8'] = 'Internal interface';
$iface_title['tun0'] = 'OpenVPN tunnel';
$vnstat_bin = '/usr/local/bin/vnstat';
$data_dir = './dumps';
$graph_format='svg';
define('GRAPH_FONT',dirname(__FILE__).'VeraBd.ttf');
define('SVG_FONT', 'Verdana');
define('DEFAULT_COLORSCHEME', 'dark');

?>
```

Создадим каталог для дампа:

```
# mkdir /usr/local/www/vnstat/dumps
```

Потом добавлем в **cron** следующие строки:

```
* /5 * * * * root /usr/local/bin/vnstat --dumpdb -i vlan8 >
/usr/local/www/vnstat/dumps/vnstat_dump_vlan8
* /5 * * * * root /usr/local/bin/vnstat --dumpdb -i tun0 >
/usr/local/www/vnstat/dumps/vnstat_dump_tun0
```

После изменений перестартовываем **cron**:

```
# killall -1 cron
```

Далее добавляем в конфигурационный блок Apache следующий блок:

```
Alias /vnstat/ "/usr/local/www/vnstat/" <Directory "/usr/local/www/vnstat/">
    Options none          AllowOverride All          Order Allow,Deny          Allow from all
</Directory>
```

После внесения изменений, необходимо перезапустить Apache:

```
# apachectl graceful
```

Несколько скриншотов:



[2]



<http://muff.kiev.ua/content/vnstat-monitorim-setevoi-trafik>

[1] <http://humdi.net/vnstat/>

[2] <http://muff.kiev.ua/files/imagepicker/1/vnstat.png>

[3] <http://muff.kiev.ua/files/imagepicker/1/vnstat1.png>