

PPPD IPv4 zone counters

Контакты

Чтоб не мучать ~Asmodeus~, вопросы и пожелания направляйте сразу мне:

- stas at onlineua.net
- 226774405

Для чего

Патч создавался для подсчёта трафика, проходящего через rppd по зонам IPv4 адресов:

- 0 - зона по умолчанию. В эту зону попадает весь трафик, который не попал в другие зоны. Данная зона не заполняется IP-диапазонами.
- 1,... - зоны в которые попадает трафик, соответствующий внесённым диапазонам.

Установка

Для работы необходимы:

1. Linux (sources)
2. rppd-2.4.2 (sources) - 2.4.3 у меня чего-то не заработал (без патча), если кто заведет, пишите.
3. Патч для linux kernel и rppd (брать с SourceForge).
4. Навыки по конфигурированию ядра Linux и наложению патчей.

Конфигурирование и сборка ядра Linux

1. Распакуйте ядро. Обычно в /usr/src/
2. Наложите патч (linux-2.6.16-ipv4zones.patch)
3. Если не нужна отладочная информация (на рабочем сервере её очень много - одна строка на каждый IP пакет на каждом из rpp), замените в drivers/net/ppp_generic.c `# define lib_log_trace(fmt, args...) printk(KERN_INFO fmt, args)` на `#define lib_log_trace(fmt, args...)`
4. Запустите конфигуратор (make menuconfig или другой)
5. Зайдите в Network device support. PPP (point-to-point protocol) support должен быть включен кому как больше нравится. В нём есть подпункт PPP IPv4 zone counters - включите, под ним есть PPP IPv4 zones number, по умолчанию 4. Если поменяете, не забудьте поправить потом в PPPD соответственно.
6. Выйдите из конфигулятора и запустите make bzImage modules etc.

Конфигурирование и сборка PPPD

1. Распакуйте rppd, наложите патчи (mpps-mppc, etc)
2. Наложите rppd-2.4.2-ipv4zones.patch

3. Поправьте CONFIG_PPP_IPV4ZONES_COUNT если поменяли в ядре. Тут ничего не могу сделать, rpppd использует локальную копию rppd_defs.h, не смотрит чего есть в kernel.
4. Если собираете сами, проверьте, чтобы были включены все необходимые plugins:
rpppd/plugins/Makefile.linux
5. Соберите и поставьте rpppd
6. Скопируйте dictionary.rpppd в sysconfdir radiusd и включите его в dictionary.

Использование

Конфигурирование IPv4 зон для подсчёта трафика

После загрузки модуля rppd_generic или загрузки ядра (если поддержка PPP вкомпилена в ядро) должен появиться файл `/proc/ipv4zones`. Если его нет, значит либо ядро собрано без патча, либо не загружен модуль. Данный файл - канал для конфигурирования зон. Для конфигурирования используются бинарные пакеты, следовательно необходима утилита. Утилиту брать на SourceForge, там же где и всё остальное. Для сборки и работы необходима библиотека libxml2, правильно прописанная в pkgconfig. Сборка осуществляется командой `make` :)

В каталоге с утилитой лежит `config.xml` - пример конфигурации. Его нужно подправить под конкретные нужды.

Для изменения конфигурации зон необходимо выполнить: `./config_parser config.xml`

!!! Внимание !!! Сейчас пока `config_parser` выводит много отладочной информации в `stdout`, ошибки выводятся в `stderr`. Но в любом случае просмотрите чего написал `config_parser` - вдруг там будет сообщение об ошибке?

Конфигурирование ABills

/ System configuration/ NAS/ Выставить тип `leppd:Linux PPPD IPv4 zone counters`

PPPD теперь будет дополнительно отправлять такие атрибуты по `accounting types Stop` и `Interim-Update` (все счётчики относительно интерфейса):

- `PPPD-Output-Packets-Zones-0` - количество пакетов, которые rpppd отправил пользователю.
- `PPPD-Input-Packets-Zones-0` - количество пакетов, которые rpppd принял от пользователя.
- `PPPD-Output-Octets-Zones-0` - количество байт, которые rpppd отправил пользователю.
- `PPPD-Input-Octets-Zones-0` - количество байт, которые rpppd принял от пользователя.

и также для остальных зон. Если зон больше 4-х, нужно поправить `dictionary.rpppd` и прописать дополнительные зоны.

!!! Внимание !!! Если у вас много зон (зон тарификации, а не разных IP адресов, которые вы включаете в зоны), лучше пользуйтесь `Ipn` - он предназначен именно для таких случаев.

Пример атрибутов, приходящих от rpppd к freeradius

```
rad_recv: Accounting-Request packet from host 192.168.254.2:2048, id=189, length=316
  Acct-Session-Id = "4490620615B3"
  User-Name = "stas"
  Acct-Status-Type = Interim-Update
  Service-Type = Framed-User
  Framed-Protocol = PPP
  Acct-Authentic = RADIUS
  Acct-Session-Time = 2225
  Acct-Output-Octets = 222
  Acct-Input-Octets = 894
  Acct-Output-Packets = 5
  Acct-Input-Packets = 13
  PPPD-Output-Packets-Zones-0 = 0
  PPPD-Input-Packets-Zones-0 = 4
  PPPD-Output-Octets-Zones-0 = 0
  PPPD-Input-Octets-Zones-0 = 328
  PPPD-Output-Packets-Zones-1 = 2
  PPPD-Input-Packets-Zones-1 = 6
  PPPD-Output-Octets-Zones-1 = 168
  PPPD-Input-Octets-Zones-1 = 492
  PPPD-Output-Packets-Zones-2 = 0
  PPPD-Input-Packets-Zones-2 = 0
  PPPD-Output-Octets-Zones-2 = 0
  PPPD-Input-Octets-Zones-2 = 0
  PPPD-Output-Packets-Zones-3 = 0
  PPPD-Input-Packets-Zones-3 = 0
  PPPD-Output-Octets-Zones-3 = 0
  PPPD-Input-Octets-Zones-3 = 0
  NAS-Port-Type = Async
  Framed-IP-Address = 192.168.253.1
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 0
  Acct-Delay-Time = 0
```

```
rad_recv: Accounting-Request packet from host 192.168.254.2:2048, id=190, length=316
  Acct-Session-Id = "4490620615B3"
  User-Name = "stas"
  Acct-Status-Type = Stop
  Service-Type = Framed-User
  Framed-Protocol = PPP
  Acct-Authentic = RADIUS
  Acct-Session-Time = 2263
  Acct-Output-Octets = 222
  Acct-Input-Octets = 894
  Acct-Output-Packets = 5
  Acct-Input-Packets = 13
  PPPD-Output-Packets-Zones-0 = 0
  PPPD-Input-Packets-Zones-0 = 4
  PPPD-Output-Octets-Zones-0 = 0
  PPPD-Input-Octets-Zones-0 = 328
  PPPD-Output-Packets-Zones-1 = 0x00000002
  PPPD-Input-Packets-Zones-1 = 0x00000006
  PPPD-Output-Octets-Zones-1 = 0x000000a8
  PPPD-Input-Octets-Zones-1 = 0x000001ec
  PPPD-Output-Packets-Zones-2 = 0
  PPPD-Input-Packets-Zones-2 = 0
  PPPD-Output-Octets-Zones-2 = 0
  PPPD-Input-Octets-Zones-2 = 0
  PPPD-Output-Packets-Zones-3 = 0
  PPPD-Input-Packets-Zones-3 = 0
  PPPD-Output-Octets-Zones-3 = 0
  PPPD-Input-Octets-Zones-3 = 0
  NAS-Port-Type = Async
  Framed-IP-Address = 192.168.253.1
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 0
```

```
| Acct-Delay-Time = 0 |
```

TODO

1. !!! Конфигуратор. Надо сделать интерфейс (ядро + программа или ядро + функция rppd) для задания зон без пересборки ядра и перезагрузки модуля.
2. ??? Per-PPP zones. Сейчас зоны общие для всех rpp.
3. ??? Конфигуратор через радиус. Задание зон при авторизации rppd на радиус-сервере.
4. ?!? Переход с отстороненного объектного интерфейса на KOBJECTs.
5. ?!! Лимит сессии по трафику в зоне.
6. !!! Переход на rbtree.c в lib оказывается есть rbtree.c, а я презевал :(
7. !!! Добавить Acct-Input-Gigawords и Acct-Output-Gigawords
8. !!! Перейти с передачи статистики по зонам в основном пакете к отдельному ioctl
9. !!! Переписать конфигуратор на основе user-space списков, убрать <linux/list.h>, так как в kernel 2.4 list наружу не экспортируется. Скорее всего на C++ std::vector.

Ideas

Возможно, лучшим выходом будет использование VSA - Vendor Specific Attributes. Например, так:

RADIUS пакет с VSA (в текстовом или бинарном виде)

- От RADIUS к NAS для задания зон будут отправляться такие атрибуты (внутри Auth-Response пакета):
 - Area=<Zone number>:<Start IP>:<Last IP>
 - Area-Speed-Limit=<Zone number>:<Upstream-Speed-Limit>:<Downstream-Speed-Limit>
- От RADIUS к NAS для ограничения будут отправляться такие атрибуты (внутри Auth-Response и/или Accounting-Response пакета):
 - Area-Session-Octets-Limit=<Zone number>:<Limit>
 - Area-Octets-Direction=<Zone number>:<Direction>
- От NAS к RADIUS будут ходить атрибуты (внутри Accounting-Request пакета):
 - Incoming-Area=<Zone number>:<Incoming packets>:<Incoming ocktets>:<Incoming Gigapackets>:<Incoming Gigawords>
 - Outgoing-Area=<Zone number>:<Outgoing packets>:<Outgoing ocktets>:<Outgoing Gigapackets>:<Outgoing Gigawords>

Встретил упоминание о POD - Packet of Disconnect. Надо посмотреть, чего можно вытянуть из RFC 3576 - Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)

ChangeLog

lepppd 0.1.2

- Исправлено неверное определение зоны при работе rppoe-server через rppoe модуль ядра (Input записывался в зону 0).
- Исправлен неверный подсчёт input traffic по всем зонам (терялось 2 байта на каждый пакет).
- Почищен вывод информации (разбивка по типу).

Configurator 0.1.1

- Переписан конфигуратор на основе std::vector. Сейчас для сборки необходим g++.
- Исправлена заглушка при получении номера зоны из config.xml. Сейчас используется atoul, без проверки на ошибки разбора.

lepppd 0.1.1

- Исправлена ошибка в размере копируемого блока.
- Максимально унифицированы патчи для linux 2.4.32 и 2.6.16.
- Конфигуратор без изменений, брать версию 0.1.0.

Changes between 0.0.1 and 0.1.0

- Выброшены файлы lib_rbtrees.[ch] - осуществлён переход на Red-Black Tree который есть в ядре.
- Добавлен конфигуратор (ppp-ipv4zones-0.1.0.tar.bz2).

From:
<http://abills.net.ua/wiki/> - **Advanced Billing Solution**

Permanent link:
<http://abills.net.ua/wiki/doku.php/abills:docs:nas:linux:lepppd:ru>

Last update: **2015/12/05 15:50**

