Table of Contents

IPFIX receiver utility 3
Introduction 3
Installation and upgrade 3
Important changes in version 1.0.3 vs 1.0.2 3
Supplied files 3
Дополнительные настройки ОС 4
Параметры запуска программы 5
Конфигурация 5
Служебные разделы 5
logger_root 6
handler_ipfixreceiverlogger 6
formatter_ipfixreceiverlogger 6
connect 7
dump 7
InfoModel 8
ExportModel 9
ExportModelFile 9
Проблемы и решения 10

IPFIX receiver utility

Introduction

IPFIX receiver is used for receiving an IPFIX (Netflow 10) stream from DPI devices and store the stream to a local file. The stored file can be processed as a text file any unix utilities.

Installation and upgrade

1. use VAS Experts repositore according to p.1 of the DPI installation instruction.
2. install ipfixreceiver:

   ```
   yum install -y ipfixreceiver
   ```

3. check changes in configuration files for installed version look at part "Important changes in version ..."

Important changes in version 1.0.3 vs 1.0.2

1. chenged configuration file in part of IP address transformation, since 1.0.3 version use decodeipv4, decodeipv6 in Export model to exoprt IP in readable mode. Expamle:

   ```
     source_ip4, decodeipv4
   ```

   ```
     destination_ip4, decodeipv4
   ```

2. saving data now in separate process, important if DPI has more than 25 000 session per second, it can load upto 2 proccesor cores. In DEBUG loging added check records to controll save processing
   (a)cnt=NNNNN - send NNNNN buffer
   (b)cnt=YYYYY - saved YYYYY buffer.
3. buffer_size parameter added - size of buffer to interchange between receiver and saver processes, use it in [dump] section, default value - 100000 records (for 20Gbe or 25 000 seesion per second). If the buffer size is not reached then 30sec timeout is used to push buffer into saver process.

Supplied files

1. configuration expamples:

   ```
   /etc/dpiui/ipfixreceiver.conf - clickstream example (http/https
   clickstream)
   /etc/dpiui/ipfixreceiverflow.conf - session example (netflow 10/IPFIX
   full session export)
   ```

```
/etc/dpiui/ipfixreceiversip.conf - meta information (sip connections)
example
```

2. programm files directory:

```
/usr/local/lib/ipfixreceiver.d/
```

3. additional files:

```
/etc/dpiui/port_proto.txt - for translation protocol number to text
protocol name
```

4. link to executable:

```
/usr/local/bin/ipfixreceiver -> link to
/usr/local/lib/ipfixreceiver.d/ipfixreceiver
```

Дополнительные настройки ОС

1. настройте iptables для приема внешних данных
   Для работы ipfixreceiver'a требуется открыть порты которые так же будут использоваться в конфигурации в разделе [connect]
   Например вами используются протокол TCP, 1500 порт и IP=212.12.11.10

   ```
   [connect]
   protocol=tcp
   host=212.12.11.10
   port=1500
   ```

   Для приема IPFIX потока у вас в /etc/sysconfig/iptables должно быть следующее правило:

   ```
   -A INPUT -p tcp -m state --state NEW -m tcp --dport 1500 -j ACCEPT
   ```

   Не забудьте что после внесения правила в iptables требуется перезапуск:

   ```
   service iptables restart
   ```

2. настройте ротацию логов
   Пример ротации для лог файла /var/log/dpiuiflow.log, создайте в директории /etc/logrotate.d/ файл flowlog следующего содержания

   ```
   /var/log/dpiui*.log {
       rotate 5
       missingok
       notifempty
       compress
       size 10M
       daily
       copytruncate
   ```

```
    nocreate
    postrotate
    endscript
}
```

Обратите внимание на использование метода copytruncate, иначе файл будет пересоздан и запись лога из процесса прекратится.
Соответственно в конфигурации ipfixreceiver у вас в разделе [handler_ipfixreceiverlogger] указано следующее:

```
args=('/var/log/dpiuiflow.log', 'a+')
```

3. Настройте удаление старых файлов. Например удаление старых архивов (более 31 дня) с записями о сессиях запакованных gzip:

```
15 4 * * * /bin/find /var/dump/dpiui/ -name url_\*.dump.gz -cmin +44640
-delete > /dev/null 2>&1
```

Измените строчку под ваши требования и добавьте в файл /var/spool/cron/root.

Параметры запуска программы

Утилита ipfixreceiver имеет следующие параметры запуска :

```
usage: ipfixreceiver start|stop|restart|status|-v [-f <config file>]
где
  start   - запуск в режиме сервиса
  stop    - останов сервиса
  state   - состояние работы сервиса
  restart - перезапуск сервиса
  -v      - вывести информацию о версии
  -f <config file> - указать файл конфигурации для запуска сервиса

Пример:
  ipfixreceiver start -f /etc/dpiui/ipfixreceiverflow.conf
```

Конфигурация

По умолчанию используется файл конфигурации /etc/dpiui/ipfixreceiver.conf .
Больше информации о конфигурировании логирования можно найти по ссылке Logging

Служебные разделы

1. loggers - определяет используемые лог идентификаторы

2. handlers - определяет используемые обработчики для сохранения лога
3. formatters - определяет используемые форматы для лога

logger_root

1. level - определяет уровень логирования (верхний уровень)
   Возможные значения:

   ```
   CRITICAL   -  только критические ошибки, минимальный уровень сообщений
   ERROR      -  включая ошибки
   WARNING    -  включая предупреждения
   INFO       -  включая информацию
   DEBUG      -  включая отладочные
   NOTSET     -  Все, максимальный уровень сообщений (включая все выше
   перечисленные)
   ```

   Пример:

   ```
   level=DEBUG
   ```

2. handlers - используемые обработчики сообщений
   Пример:

   ```
   handlers=ipfixreceiverlogger
   ```

handler_ipfixreceiverlogger

1. class - класс обработчика
   Пример:

   ```
   class=FileHandler
   ```

2. level - уровень сообщений

   ```
   level=DEBUG
   ```

3. formatter - наименование формата сообщений

   ```
   formatter=ipfixreceiverlogger
   ```

4. args - параметры обработчика

   ```
   args=('/var/log/dpiuiflow.log', 'a+')
   ```

formatter_ipfixreceiverlogger

1. format - описание формата сообщения

Пример:

```
format=%(asctime)s - %(name)s - %(levelname)s - %(message)s
где
%(name)s       -  имя лога
%(levelname)s  -  уровень сообщения  ('DEBUG', 'INFO', 'WARNING', 'ERROR',
'CRITICAL').
%(asctime)s    -  дата, по умолчанию формат “2003-07-08 16:49:45,896” (поле
запятаой указаны миллисекунды).
%(message)s    -  сообщение
```

2. datefmt - описание формата даты
   Пример:

```
datefmt='%m-%d %H:%M'
```

connect

1. protocol - протокол (tcp или udp).

```
protocol=udp
```

2. host - IP или имя сервера.

```
host=localhost
```

3. port - номер порта.

```
port=9996
```

dump

1. rotate_minutes - период в минутах, по прошествии которого временный файл в dumpfiledir/<port>.url.dump будет перемещен в архив (mv) и создан новый временный файл.

```
rotate_minutes=10
```

2. processcmd - команда которая будет запущена по окончании ротации файла, параметр имя файла с путем к нему.

```
processcmd=gzip %%s
```

3. dumpfiledir - директория куда будут сохраняться файлы с принятыми данными.

```
dumpfiledir=/var/dump/dpiui/ipfixflow/
```

4. buffer_size - размер буфера обмена между процессом приема и записи в файл, по умолчанию значение параметра 100000 записей (ориентировано на 20Гбит трафика или 25 000 сессий в сек). Если к-во сессий в секунду значительно меньше, то обязательно

пропорционально измените данный параметр.

InfoModel

Блок описывает получаемые данные по IPFIX протоколу.

1. InfoElements - параметр с описанием элементов информационной модели для IPFIX

```
InfoElements =  octetDeltaCount,       0,    1, UINT64, True
                packetDeltaCount,      0,    2, UINT64, True
                protocolIdentifier,    0,    3, UINT8
                session_id,        43823, 2000, UINT64, True
где,
  session_id - наименование поля из описания IPFIX см. разделы
 43823 - уникальный номер организации (enterprise number)
 1    - уникальный номер поля
  UINT64 - тип поля
  True   - использовать обратный порядок байт (endian). Значения - True или
пусто.
```

Типы полей:

Type	Length	Type IPFIX
OCTET_ARRAY	VARLEN	octetArray
UINT8	1	unsigned8
UINT16	2	unsigned16
UINT32	4	unsigned32
UINT64	8	unsigned64
INT8	1	signed8
INT16	2	signed16
INT32	4	signed32
INT64	8	signed64
FLOAT32	4	float32
FLOAT64	8	float64
BOOL	1	boolean
MAC_ADDR	6	macAddress
STRING	VARLEN	string
SECONDS	4	dateTimeSeconds
MILLISECONDS	8	dateTimeMilliseconds
MICROSECONDS	8	dateTimeMicroseconds
NANOSECONDS	8	dateTimeNanoseconds
IP4ADDR	4	ipv4Address
IP6ADDR	16	ipv6Address

Наименование полей и описание можно взять по ссылкам:

1. Шаблон экспорта Netflow в формате IPFIX
2. Шаблоны экспорта clickstream и SIP

Дополнительная информация:
Information Model for IP Flow Information Export

ExportModel

определяет параметры модели для экспорта, зарезервировано для будущего использования.

1. Mode - тип используемого экспорта

```
Mode = File
```

ExportModelFile

Описание модели экспорта File.

1. Delimiter разделитель полей в строке (\t - табуляция, еще примеры - |,;)

```
Delimiter = \t
```

2. ExportElements - описание полей которые будут сохранены в файл.

```
ExportElements = timestamp, seconds, %%Y-%%m-%%d %%H:%%M:%%S.000+03
                 login
                 source_ip4
                 destination_ip4
                 host, decodehost
                 path, decodepath
                 referal, decodereferer
                 session_id
где поля в каждой строке:
 имя - наименование поля из информационной модели [InfoModel] (login,
session_id и т.п.)
 обработчик - процедура обработки поля перед выводом
                seconds        -  поле в секундах, ожидается формат
                milliseconds   -  поле в милисекундах, микросекундах,
наносекундах ожидается формат
                decodehost     -  перекодировать из punycode в UTF-8
                decodepath     -  перекодировать из urlencoding в UTF-8
                decodereferer  -  перекодировать из (punycode,urlencoding) в
UTF-8
                decodeproto    -  перекодировать идентификатор протокола в
строку
 формат - описание формата для seconds, milliseconds.
      Пример: %%Y-%%m-%%d %%H:%%M:%%S.%%f+0300
      Результат: 2016-05-25 13:13:35.621000+0300
```

Проблемы и решения

1. как получить версию утилиты?
   Используйте следующие команды:

   ```
   ipfixreceiver -v
   ```

   ```
   yum info ipfixreceiver
   ```

2. можно ли на один порт отправлять IPFIX потоки с разных DPI?
   Да. Единственное в записываемом потоке их будет не различить.
3. как понять, что утилита работает?
   a) проверьте, что порт из конфигурации прослушивается утилитой, например 1500:

   ```
   netstat -nlp | grep 1500
   ```

   b) проверьте лог, нет ли ошибок
   c) Проверьте, что запись в промежуточный файл происходит, например для 9996 порта (директория для файлов - /var/dump/dpiui/ipfixurl):

   ```
   tail -f /var/dump/dpiui/ipfixurl/9996.url.dump
   ```

4. все проверено, но приема сообщений нет?
   a) забыли открыть порт в iptables.
   b) инициализировали ipfixreceiver с неверным IP сервера.
5. с DPI идет большое количество сессий (более 2 млн сессий/мин), при включенном DEBUG режиме видно, что счетчик обмена буферами не успевает записать до получения следующего блока записей, что можно сделать?
   a) удалите преобразование даты в строку, это уменьшит процессорное время на обработку и дополнительно получите уменьшение объема результирующего файла
   b) удалите преобразование decodeipv4, не значительно, но так же получите ускорение записи файла
   c)настройте buffer_size при к-ве сес /сек более 30к совместно с п.d
   d) увеличьте частоту процессора и объем памяти