Table of Contents

ipfixreceiver2: IPFIX/NetflowV9 collector

Introduction

ipfixreceiver2 is an IPFIX/NetflowV9 collector with the following functionality

- Allows to save the received data with the required format in a text file.
- Allows to redirect received data to other IPFIX collectors.

Installation and Upgrading

CentOS

1. Add the VAS Experts repository

```
rpm --import http://vasexperts.ru/centos/RPM-GPG-KEY-vasexperts.ru
rpm -Uvh
http://vasexperts.ru/centos/6/x86_64/vasexperts-repo-1-0.noarch.rpm
```

2. Add the EPEL repository
3. Install the ipfixreceiver2:

```
yum install -y ipfixreceiver2
```

4. In order to upgrade ipfixreceiver2 issue the following command:

```
yum update -y ipfixreceiver2
```

ipfixreceiver2 files

- Files describing the types of ipfix data fields:

```
/etc/rcollector/xml/ipfix_raw.xml - ipfix data field types used in
fullflow.
/etc/rcollector/xml/ipfix_url.xml - ipfix data field types used in
clickstream (http requests).
/etc/rcollector/xml/ipfix_sip.xml - ipfix data field types used in SIP
connections.
/etc/rcollector/xml/ipfix_aaa.xml - ipfix data field types used in AAA
events.
/etc/rcollector/xml/ipfix_nat.xml - ipfix data field types used in NAT
events.
```

- Examples of configuration files describing ipfix data import and export models:

```
/etc/rcollector/ipfixreceiver_raw.ini is responsible for ipfix data
import and export for fullflow.
/etc/rcollector/ipfixreceiver_raw_new.ini is responsible for ipfix data
import and export for the VAS Experts DPI version 8.1 and higher.
/etc/rcollector/ipfixreceiver_url.ini is responsible for ipfix data
import and export for clickstream.
/etc/rcollector/ipfixreceiver_sip.ini is responsible for ipfix data
import and export for SIP connections.
/etc/rcollector/ipfixreceiver_aaa.ini is responsible for ipfix data
import and export for AAA events.
/etc/rcollector/ipfixreceiver_nat.ini is responsible for ipfix data
import and export for NAT events.
```

- Executable file,:

```
/usr/bin/ipfixreceiver2
```

CentOS 6

- Scripts used to start the process of importing and exporting ipfix data:

```
/etc/init.d/ipfix_raw - ipfixreceiver2 startup script with
corresponding /etc/rcollector/ipfixreceiver_raw.ini configuration file.
/etc/init.d/ipfix_url - ipfixreceiver2 startup script with
corresponding /etc/rcollector/ipfixreceiver_url.ini configuration file.
/etc/init.d/ipfix_sip - ipfixreceiver2 startup script with
corresponding /etc/rcollector/ipfixreceiver_sip.ini configuration file.
/etc/init.d/ipfix_aaa - ipfixreceiver2 startup script with
corresponding /etc/rcollector/ipfixreceiver_aaa.ini configuration file.
```

CentOS 7

- systemd-based configuration files (systemd units) to start the process of importing and exporting ipfix data:

```
/usr/lib/systemd/system/ipfix_raw.service - systemd unit responsible
for starting ipfixreceiver2 with corresponding
/etc/rcollector/ipfixreceiver_raw.ini configuration file.
/usr/lib/systemd/system/ipfix_url.service - systemd unit responsible
for starting ipfixreceiver2 with corresponding
/etc/rcollector/ipfixreceiver_url.ini configuration file.
/usr/lib/systemd/system/ipfix_sip.service - systemd unit responsible
for starting ipfixreceiver2 with corresponding
/etc/rcollector/ipfixreceiver_sip.ini configuration file.
/usr/lib/systemd/system/ipfix_aaa.service - systemd unit responsible
for starting ipfixreceiver2 with corresponding
/etc/rcollector/ipfixreceiver_aaa.ini configuration file.
```

ipfixreceiver2 startup options

ipfixreceiver2 utility has the following startup options:

```
usage: ipfixreceiver2 <-f config file> [options]
here
--daemon                    start the program as a daemon process.
--umask=mask                set umask (octal value, 027 is the default one).
--pidfile=path              set path to a pid file.
-h, --help                  display a brief description.
-fFILE, --config-file=FILE  set path to the configuration file.
-v, --version               display program version.
```

Configuration

Configuration options are specified in the .ini file.

Section [connect]

The section is used to specify the parameters for receiving ipfix data.

- protocol - IP protocol (tcp or udp)
  
  Before using the udp protocol, you should make sure that the size of the ipfix record does not exceed the size of the MTU (clickstream data can be received using the tcp protocol only).
- host - interface, used to receive the data
- port - port number
- flow_type - the type of flow to receive: ipfix or netflow9. When netflow9 protocol is used the flow_type can be equal to 'udp' only.

Section [dump]

The section is used to specify the parameters of data dump received.

- delimiter - character used as delimiter within the file.
- rotate_minutes - the time period after which the temporary file will be closed and renamed to a persistent one
- rotate_flows - the ipfix records number upon the exceeding of which the temporary file will be closed and renamed to a persistent one. 0 - to disable this rotation type.
- dumpfiledir - directory used to store the dump files.
- fileprefix - dump file name prefix.
- rotateformat - generates a dump file name.
- extension - extension of a dump file.
- temp_file_suffix - name suffix of a temporary file.
- processcmd - command used to set rotate option. %s specifies the name of persistent file

containing the dump.
- detach_child - when is set to true, the processcmd process will be unbound from ipfixreceiver's process.
- decode_url - to decode characters in url when using decodepath.
- decode_host - to decode idna within the host name when using decodehost.
- decode_referer - to decode idna to referer when using decodereferer.
- reopen_time - the time period upon the exceeding of which the attempt to reopen a file for recording a dump after an error occurred (when attempting to access the file) will be made. The default value is 30 seconds.
- checkdir - boolean parameter; is used to check whether dumpfiledir exists and, if it does not exist, the corresponding directories will be created (including all the dumpfiledir subdirectories). The default value is true.
- fw_max_elements_in_queue - the items number upon the exceeding of which they will be forwarded to the queue to be written to the file. The default value is 100000.
- fw_max_queue_size - the maximum number of arrays of elements in the queue. If the number of people at the time of adding them to the queue will be more than fw_max_queue_size, then the data will be discarded. The default value is 2.
- bad_characters - characters that do not need to be displayed when writing to a file. Single characters along with escape sequences can be specified. Default value is “\t\r\n;\x00”.

Section [InfoModel]

This section specifies an xml file describing the type of data within the received ipfix flow.

- XMLElements - path to xml file with data type description using the IANA IPFIX Entities registry format.

Section [Template]

The section is responsible for the data sequence order within the received ipfix flow and, if necessary, for the received data filtering by the given identifier.

- Elements - comma separated list of received data.
- filter_tid - only the data with the given identifier will be processed, all the other ones will be discarded.

Section [ExportModel]

This section specifies the order and format of the received data to be exported.

- Elements - comma separated list of data to be stored in the file. You can change the predefined export format (the data will be exported to a file) for each data type using the following format: field_name: output_format [: option]. The following types of data output are possible:

Output_format	**Description**
decode_unsigned	Decode as unsinged
decode_signed	Decode as signed
decodeipv4	Decode as IPv4 address

Output_format	Description
decodeipv6	Decode as IPv6 address
decode_string	Decode as string
decode_seconds	Decode as date and time in seconds. The default output format is '%Y-%m-%d %H:%M:%S'. You can specify date/time format on your own.
decode_milliseconds	Decode as date and time in milliseconds. The default output format is '%Y-%m-%d %H:%M:%S'. You can specify date/time format on your own.
decodehost	Decode as host name
decodepath	Decode as url
decodereferer	Decode as referer

Section [stats]

The section specifies the export options for sending ipfixreceiver2 statistics (metrics and events) to the telegraf agent.

- socket_path - path to the telegraf's datagram socket.
- interval - the time period upon the exceeding of which the statistics will be sent to the telegraf agent.
- tag - tag set in the ipfix_tag field when sending statistics to the telegraf agent.

Section [export]

- to - specifies the collector addresses to be used to export received ipfix records.
  Used format: ip/port/proto[,ip/port/proto]. For example:

```
[export]
to=10.0.0.2/9921/tcp, 10.0.0.3/3444/udp
```

 When using the udp protocol, you should make sure that one ipfix record does not exceed the size of the MTU.

Section [logging]

The section specifies the logging parameters.

- loggers.root.level - log level
- loggers.root.channel - channel to display messages
- channels.fileChannel.class - output channel class
- channels.fileChannel.path - path to the log file
- channels.fileChannel.rotation - rotation parameter
- channels.fileChannel.archive - archive file name parameter
- channels.fileChannel.purgeCount - number of archive files
- channels.fileChannel.formatter.class - formatter class
- channels.fileChannel.formatter.pattern - formatter pattern
- channels.fileChannel.formatter.times - time

For more information about logging parameters please follow the Class FileChannel link.

Configuration examples

Receiving of ipfix data

The /etc/rcollector/ipfixreceiver_*.ini files provide configuration examples for receiving various ipfix data flows. Before starting the program, you should to change the configuration file to meet your requirements.

- If necessary, make changes to the [connect] section, specifying the interface, port and protocol for receiving ipfix data.
- Specify within [dump] section the following stuff:
  - dumpfiledir - the directory where the temporary file and data files will be created.
  - rotate_minutes - the time period upon the exceeding of which to close the temporary file, rename it to a file with a permanent name and execute a command from the processcmd parameter to operate on the received file.
  - processcmd - this command should be executed on the file with data.
  - delimiter - delimiter character between data fields.
- You should specify the required order of the fields in the saved file within the [ExportModel] section.

Exporting of ipfix data

To export the data received by ipfix, you need to make changes to the configuration file by adding the [export] section and specifying the destination addresses. For example, to send ipfix data to an ipfix collector having the 10.0.0.5:1501 address using the tcp protocol, the configuration item within [export] section will look like this:

```
[export]
to = 10.0.0.5/1501/tcp
```

If you need to specify multiple ipfix collectors, you can specify comma-separated list of ipfix collectors. For example:

```
[export]
to = 10.0.0.5/1501/tcp, 192.168.1.200/1501/tcp
```