NETFLOW/IPFIX

Teldat Dm789-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

Teldat Dm702-I TCP-IP Configuration

Teldat Dm786-I AFS

Chapter 1 Introduction

1.1 NETFLOW/IPFIX Protocol: Description

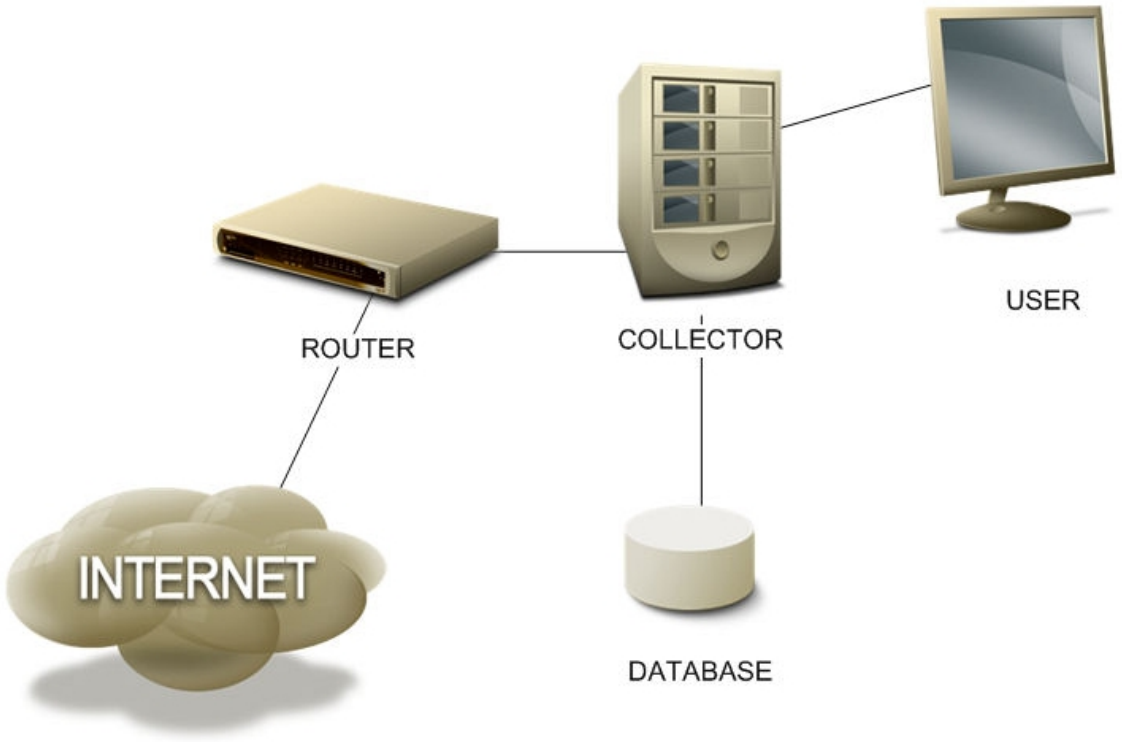
1.1.1 Definition

The NETFLOW/IPFIX protocol allows you to monitor data flows in an IP network.

This simply lets you see when, where and what traffic flows are being processed by the network and who is doing said processing. By understanding the behavior of traffic flows, an IP network can be improved and its actions can be better accounted for.

NETFLOW/IPFIX exports IP flow information in packets, encapsulated in UDP, using a certain format. This format depends on the configured version, meaning our devices can use NETFLOW version 5, version 9 or IPFIX (Netflow version 10). Optionally, these packets can be encrypted using the DTLS (Datagram Transport Layer Security) protocol before being sent.

UDP packets are received by a *collector* server, which interprets and stores flows in a database. Subsequently, the network administrator can check said database to obtain graphics and statistics on traffic processed by the router.



Version 5 NETFLOW packet format follows a predefined pattern so the same information is always sent. NETFLOW version 9 and IPFIX, however, are more flexible regarding the information sent in each flow. Since these formats are flexible, information on the specific fields transmitted is sent periodically. This field definition is known as a *template*. Details on both protocols can be found in IETF RFCs 3954 and 7011.

Regardless of the flow exporting protocol and version, the router internally builds a flow cache. Each flow is a unidirectional group of packets that share the following parameters:

(1) Source IP address.

(2) Destination IP address.

(3) IP protocol.

(4) Source port (for UDP/TCP/SCTP protocols or type/code for ICMP protocol).

(5) Destination port (for UDP/TCP/SCTP protocols).

(6) IP header TOS field.

(7) Input interface SNMP Index.

(8) Output interface SNMP Index.

If any of these parameters vary, the packet is considered to belong to a different flow.

A packet is processed by the NETFLOW/IPFIX subsystem if it enters the router through an interface where the **ip flow ingress** command is enabled, or if it is forwarded by an interface where the **ip flow egress** command is enabled. Each packet is only registered once per VRF, meaning it cannot be considered an ingress and an egress simultaneously. In this case, flow direction shall be ingress.

When a packet is processed by the NETFLOW/IPFIX subsystem, a flow match in the cache is searched for. If there is a match, the flow is updated (increasing packet and byte counters) and the lifetime is refreshed. If there is no matching flow, a new one is created and inserted into the cache.

The router exports a flow record when it determines said flow has expired and is deleted from the cache. A flow is considered expired when, during a certain period of time (15 seconds by default), no packets associated to said flow are routed. A flow is also considered finished when it has been active for a long time (default 30 minutes, although 1 minute is recommended for best resolution and lowest delay).

Exported NETFLOW/IPFIX flow records contain the following fields:

Field	Description	NF V5	NF V9	IPFIX	ID	SIZE
sourceIPv4Address	IPv4 source address.	Always	Always	Always	8	4
destinationIPv4Address	IPv4 destination address.	Always	Always	Always	12	4
ipNextHopIPv4Address	IPv4 next-hop address.	Always	Always	Always	15	4
ingressInterface	Input interface SNMP ifIndex.	Always	Always	Always	10	2
egressInterface	Output interface SNMP ifIndex.	Always	Always	Always	14	2
packetDeltaCount	Number of packets.	Always	Always	Always	2	4
octetDeltaCount	Number of bytes.	Always	Always	Always	1	4
flowStartSysUpTime	System uptime in milliseconds for first packet. Command: **ip export time-format uptime** (default).	Always	Default	Default	22	4
flowEndSysUpTime	System uptime in milliseconds for last packet. Command: **ip export time-format uptime** (default).	Always	Default	Default	21	4
flowStartSeconds	Absolute timestamp in seconds for first packet. Command: **ip export time-format seconds**.	No	Opt.	Opt.	150	4
flowEndSeconds	Absolute timestamp in seconds for last packet. Command: **ip export time-format seconds**.	No	Opt.	Opt.	151	4
flowStartMilliseconds	Absolute timestamp in milliseconds for first packet. Command: i**p export time-format milliseconds**.	No	Opt.	Opt.	152	8
flowEndMilliseconds	Absolute timestamp in milliseconds for last packet. Command: **ip export time-format milliseconds**.	No	Opt.	Opt.	153	8
sourceTransportPort	TCP/UDP/SCTP source port number.	Always	Always	Always	7	2
destinationTransportPort	TCP/UDP/SCTP source port number. ICMP type and code.	Always	Always	Always	11	2
tcpControlBits	TCP flags seen for this flow.	Always	Always	Always	6	1
protocolIdentifier	IP protocol byte.	Always	Always	Always	4	1
ipClassOfService	IP TOS byte.	Always	Always	Always	5	1
bgpSourceAsNumber	0 is always sent in V5.	Always	No	No	16	2
bgpDestinationAsNumber	0 is always sent in V5.	Always	No	No	17	2
sourceIPv4PrefixLength	Source address mask in slash notation.	Always	Always	Always	9	1
destinationIPv4PrefixLength	Destination address mask in slash notation.	Always	Always	Always	13	1

Field	Description	NF V5	NF V9	IPFIX	ID	SIZE
flowDirection	Flow direction relative to the observation point: 0 - ingress 1 - egress.	No	Always	Always	61	1
applicationId	Application ID as defined in RFC6759. Command: **collect app-id**.	No	Opt.	Opt.	95	4
selectorId	Policy Based Routing AFS session-mark value. Command: **collect session-mark.**	No	Opt.	Opt.	302	4
subApplicationId	AFS app-detect extracted fields. Commands: **collect http-host / http-referer / http-url / http-user-agent / ssl-server**.	No	No	Opt.	45003	n

The IPFIX subApplicationId information element is a variable size field that can be used to export different flow information using the following format specification:

Element Name	AppID (4 Bytes)	SubAppID (2 Bytes)	String Value (Variable lenth)	Command
sslCommonName	0x0D0001C5	0x3401	SSL Server Name Identifier	**collect ssl-server**
httpUrl	0x03000050	0x3401	HTTP Url	**collect http-url**
httpHostName	0x03000050	0x3402	HTTP Host	**collect http-host**
httpUserAgent	0x03000050	0x3403	HTTP User Agent	**collect http-user-agent**
httpReferer	0x03000050	0x3404	HTTP Referer	**collect http-referer**

1.1.2 Relationship with other subsystems

IPSEC:

If the router is applying IPSEC to the flow, packet match is performed before IPSEC encapsulation for outgoing packets and after IPSEC decapsulation for incoming packets.

Versions prior to 11.01.01 generated two flows when performing IPSEC (with and without encapsulation).

NAT:

Flow packet match is done before applying outbound NAT to the source IP, and inbound NAT to the destination IP, so that flows have local IP addresses.

Access control:

The packets dropped by an access-group, either at input or output (please see manual Teldat Dm702-I TCP-IP Configuration), are always considered ingress traffic (never egress).

IP fragmentation:

Only the first IP fragment belonging to a packet is matched against the flow cache, as it's the only one that has a UDP/TCP header. The following fragments are not processed unless the AFS system is active (please see manual Teldat Dm786-I AFS). Since the system incorporates an advanced defragmenter, no IP fragmented packets reach the netflow protocol and all are processed normally.

VRF:

Flows have unique input and output interfaces. This means that, if a packet traverses multiple VRFs with different interfaces associated to them, the packet will generate flows in every VRF that has interfaces where flow accounting is enabled.

GRE:

If the router is forwarding packets with a GRE tunneling interface, there are two different packet headers for a flow: one for the encapsulated GRE flow tunneled between the router and the other endpoint, and one for the original flow between the external endpoints. Depending on what interfaces are used for flow accounting, the flow cache displays one or both flows.

Chapter 2 Configuring NETFLOW/IPFIX

2.1 Global NETFLOW/IPFIX configuration

To configure NETFLOW/IPFIX, enter **feature netflow** from the main configuration menu.

Syntax:

```
Config$feature netflow

-- NETFLOW/IPFIX Configuration --

NETFLOW config$
```

2.1.1 [NO] COLLECT

Commands linked to the configuration of NETFLOW 9 and IPFIX flow record fields.

2.1.1.1 [NO] COLLECT APP-ID

Includes the application ID in the NETFLOW V9/IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled to generate the application ID (please see manual Teldat Dm786-I AFS).

Syntax:

```
NETFLOW config$collect app-id
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.1.2 [NO] COLLECT HTTP-HOST

Includes the HTTP Host header hostname in the IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled, as well as the HTTP Host detector (**app-detect http host**command, please see manual Teldat Dm786-I AFS). If the **not-found-txt <word>** option is configured, HTTP detected flows (where no Host header is detected) are exported with the configured text as hostname. Otherwise, an empty hostname is exported for said flows.

Syntax:

```
NETFLOW config$collect http-host ?
  <cr>
  not-found-txt <word>   Domain name to use for HTTP flows with no host detected
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.1.3 [NO] COLLECT HTTP-REFERER

Includes the HTTP Referer in the IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled, as well as the HTTP Referer detector (**app-detect http referer** command, please see manual Teldat Dm786-I AFS). If the **not-found-txt <word>** option is configured, HTTP detected flows where no Referer is found are exported with the configured text as Referer. Otherwise, an empty Referer is exported in said flows.

Syntax:

```
NETFLOW config$collect http-referer ?
  <cr>
  not-found-txt <word>   Referer to use for HTTP flows with no referer detected
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.1.4 [NO] COLLECT HTTP-URL

Includes the HTTP URL in the IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled, as well as the HTTP URL detector (**app-detect http url** command, please see manual Teldat Dm786-I AFS).

Syntax:

```
NETFLOW config$collect http-url
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.1.5 [NO] COLLECT HTTP-USER-AGENT

Includes the HTTP User-Agent in the IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled, as well as the HTTP User-Agent detector (**app-detect http user-agent** command, please see manual Teldat Dm786-I AFS). If the **not-found-txt <word>** option is configured, HTTP detected flows where no User-Agent is found are exported with the configured text as User-Agent. Otherwise, an empty User-Agent is exported for said flows.

Syntax:

```
NETFLOW config$collect http-user-agent ?
  <cr>
  not-found-txt <word>   User-agent to use for HTTP flows with no user-agent detected
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.1.6 [NO] COLLECT SESSION-MARK

Includes the AFS session-mark value set by a policy route-map in the NETFLOW V9/IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled (please see manual Teldat Dm786-I AFS).

Syntax:

```
NETFLOW config$collect session-mark
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.1.7 [NO] COLLECT SSL-SERVER

Includes the SSL Server name in the IPFIX-exported flow records. The Advanced Firewalling System (AFS) must be enabled, as well as the SSL Host detector (**app-detect ssl host** command, please see manual Teldat Dm786-I AFS). If the **not-found-txt <word>** option is configured, SSL detected flows where no Server name is found are exported with the configured text as Server name. Otherwise, an empty Server name is exported for these flows.

Syntax:

```
NETFLOW config$collect ssl-server ?
  <cr>
  not-found-txt <word>   Domain name to use for SSL flows with no host detected
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.2 [NO] IP

2.1.2.1 [NO] IP CACHE ENTRIES

Configures the maximum number of flows that can be created. Default is 65536.

If the maximum number of flows is surpassed, then the oldest are eliminated and exported.

Syntax:

```
NETFLOW config$ip cache entries ?
  <1..524288>    Value in the specified range
```

2.1.2.2 [NO] IP CACHE TIMEOUT ACTIVE

Configures the maximum lifetime that an active flow can remain in the cache before being deleted and the information exported. Default is 30 minutes.

For best granularity and lower report delay, we recommend using 1 minute flow lifetime. Having said this, the NETFLOW/IPFIX exporting bandwidth increase should be taken into account.

Syntax:

```
NETFLOW config$ip cache timeout active ?
  <1m..1h>    Time value
```

2.1.2.3 [NO] IP CACHE TIMEOUT INACTIVE

Configures the maximum time an inactive flow can remain in the cache before being deleted and exported. Default is 15 seconds (i.e. if the number of packets registered in a flow does not increase over a 15-second period, the flow is deleted and exported).

Syntax:

```
NETFLOW config$ip cache timeout inactive ?
  <1s..10m>    Time value
```

2.1.2.4 [NO] IP EXPORT DESTINATION

Configures the device so that it exports the flows to the configured IP, UDP port, VRF and DTLS. Exporting the flows to the port, DTLS and VRF is optional (port 9996 should be used). However, unless otherwise specified, flows are exported to the main VRF and UDP port by default. If you choose the DTLS encryption option, DTLS port 4740 is used by default. Also, DTLS transport can be configured for the active VRF. The device will send copies of the packets to as many destinations as you wish to define.

Syntax:

```
NETFLOW config$ip export destination ?
  <a.b.c.d>    Ipv4 format
     <1..65535>    Server port value
     vrf <1..32 chars>    VPN Routing/Forwarding instance name
        transport-dtls    Datagram Transport Layer Security
           <1..65535>    Server port value
           <cr>
     transport-dtls    Datagram Transport Layer Security
        <1..65535>    Server port value
        <cr>
     <cr>
```

Command history:

Version	Modification
11.01.06	The "transport-dtls" option was introduced as of version 11.01.06.

2.1.2.5 [NO] IP EXPORT HTTP-REFERER-AS-HOST

If the **collect http-referer** command is configured, the router sends referer information using the subApplicationID for httpHostName (13314) instead of httpReferer (13316). This command may be useful to send the flow HTTP Referer, instead of the Host, to a collector that only supports httpHostName subApplicationID. If the **collect http-referer**, **collect http-host** and **ip export http-referer-as-host** commands are all configured, only one httpHostName subApplicationId field is sent with the referer information.

Syntax:

```
NETFLOW config$ip export http-referer-as-host
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.2.6 [NO] IP EXPORT ID

Configures the NETFLOW V5 EngineID, NETFLOW V9 Source ID or IPFIX Observation Domain ID sent in the header of all exported packets. By default, 0 is exported in these fields.

Syntax:

```
NETFLOW config$ip export id ?
  <0..4294967295>    Value in the specified range
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.2.7 [NO] IP EXPORT INTERFACE-TABLE

Exports a NETFLOW V9 / IPFIX Option Template (ID 256) with the list of router interfaces containing their names and description mappings.

By default, Interface table Option Templates are exported every 10 minutes or according to the value configured through **ip export template timeout**.

The following fields are exported in each data record:

Field	Description	NF V9	IPFIX	ID	SIZE
ingressInterface	Interface SNMP ifIndex.	Always	Always	10	2
interfaceName	Interface Name.	Always	Always	82	20
interfaceDescription	Number of packets.	Always	Always	83	64

Syntax:

```
NETFLOW config$ip export interface-table
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.2.8 [NO] IP EXPORT PACKET-SIZE

Configures the NETFLOW/IPFIX maximum packet size. The packet size configured is the maximum UDP payload of the packets in bytes. Default is 1400 bytes.

Syntax:

```
NETFLOW config$ip export packet-size ?
  <512..1500>    Value in the specified range
```

Command history:

Version	Modification

11.01.01	This command was introduced as of version 11.01.01. Before this command was implemented, maximum packet size was set to 512.

2.1.2.9 [NO] IP EXPORT SERIAL-NUMBER

Configures the sending of the router serial number via the NetFlow9/IPFIX Option Template.

Syntax:

```
NETFLOW config$ip export serial-number ?
  <CR>
```

Command history:

Version	Modification
11.01.02	This command was introduced as of version 11.01.02.

2.1.2.10 [NO] IP EXPORT SESSION-MARK

Configures a name associated to an AFS session mark set by a policy route-map. The name, along with the session-mark value, is exported as an Option Template. The collector can set flows as belonging to a type of traffic identified by a name.

Syntax:

```
NETFLOW config$ip export session-mark ?
  <1..65536>    Value in the specified range
  <1..256 chars>    Name associated to mark
```

Command history:

Version	Modification
11.01.02	This command was introduced as of version 11.01.02.

2.1.2.11 [NO] IP EXPORT SOURCE

Configures the IP used as source in the netflow UDP packets sent by the device. If this is not configured, then the internal IP is used.

Syntax:

```
NETFLOW config$ip export source ?
  <a.b.c.d>    Ipv4 format
```

2.1.2.12 [NO] IP EXPORT SSL-HOST-AS-HTTP

If **collect ssl-server** is configured, SSL Host information is sent using the subApplicationID for httpHostName (13314) instead of sslCommonName (13313). This command may be useful to send the SSL Host to a collector that only supports httpHostName subApplicationID or to mix both HTTP and SSL hosts in the same flow information field.

Syntax:

```
NETFLOW config$ip export ssl-host-as-http
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.2.13 [NO] IP EXPORT TEMPLATE REFRESH-RATE

Configures the number of flow packets needed before sending the template where the fields definition is found. This only applies to NETFLOW V9 and IPFIX. Default is 20 packets.

Syntax:

```
NETFLOW config$ip export template refresh-rate ?
  <1..600>    Value in the specified range
```

2.1.2.14 [NO] IP EXPORT TEMPLATE TIMEOUT

Configures the maximum time that can lapse without sending the template where the fields definition is found. This only applies to NETFLOW V9 and IPFIX. Default is 10 minutes.

Syntax:

```
NETFLOW config$ip export template timeout ?
  <1m..2d12h>    Time value
```

Command history:

Version	Modification
11.01.01	The default value has been changed as of version 11.01.01. In previous versions, it was 30 minutes.

2.1.2.15 [NO] IP EXPORT TIME-FORMAT

Configures NETFLOW 9/IPFIX time formats for the first/last packet timestamps of the flow. By default, the traditional NETFLOW flowStartSysUpTime/flowEndSysUpTime IDs based on router uptime are sent in milliseconds. Absolute flow start/end in seconds and absolute flow start/end in milliseconds (64 bits) can be configured.

Syntax:

```
NETFLOW config$$ip export time-format ?
  uptime          Uptime relative flow start/end in milliseconds
  seconds         Absolute flow start/end in seconds
  milliseconds    Absolute flow start/end in milliseconds
```

Command history:

Version	Modification
11.01.01	This command was introduced as of version 11.01.01.

2.1.2.16 [NO] IP EXPORT VERSION

Configures which NETFLOW/IPFIX version is used. No version is used by default and this feature is disabled. To enable NETFLOW/IPFIX, at least one interface **ip flow ingress/egress** command must be configured.

Syntax:

```
NETFLOW config$ip export version ?
  5        Netflow version 5
  9        Netflow version 9
  ipfix    Ipfix (Netflow version 10)
```

Command history:

Version	Modification
11.01.01	The IPFIX version was introduced as of version 11.01.01.

2.1.3 [NO] MODE

2.1.3.1 [NO] MODE RANDOM ONE-OUT-OF

When this option is configured, one out of every *n* IP packets is processed by netflow. The number of packets and bytes in each flow is multiplied by *n* to compensate the sample. This means CPU resources are saved but packet and byte values become estimates.

Syntax:

```
NETFLOW config$mode random one-out-of ?
  <1..65535>    Value in the specified range
```

2.2 Interface NETFLOW/IPFIX configuration

Accesses the configuration menu for an interface. For example, interface ethernet0/0:

```
Config$network ethernet0/0

-- Ethernet Interface User Configuration --
ethernet0/0 config$
```

The available commands are:

2.2.1 [NO] IP FLOW INGRESS

Enables NETFLOW/IPFIX processing for IP packets entering through the interface.

You can also specify an access list so that the only incoming IP packets processed are those that match the access list.

```
ifc config$ip flow ingress ?
  list    Access-list to select traffic to be accounted
     <1..1999>    Value in the specified range
  <cr>
```

2.2.2 [NO] IP FLOW EGRESS

Enables NETFLOW/IPFIX processing for IP packets leaving through the interface.

You can also specify an access list so that the only outgoing IP packets processed are those that match the access list.

```
ifc config$ip flow egress ?
  list    Access-list to select traffic to be accounted
     <1..1999>    Value in the specified range
  <cr>
```

Chapter 3 Monitoring NETFLOW/IPFIX

3.1 Monitoring NETFLOW/IPFIX

The NETFLOW/IPFIX monitoring commands must be entered at the NETFLOW/IPFIX monitoring menu. To access this menu, use the **feature netflow** command found at the general monitoring menu.

```
+feature netflow

-- NETFLOW/IPFIX Monitor --

NETFLOW Mon+
```

3.1.1 CLEAR

3.1.1.1 CLEAR CACHE

Deletes the flows present in the cache. Flows cleared through this command are not exported.

Syntax:

```
NETFLOW Mon+clear cache
NETFLOW Mon+
```

3.1.2 LIST

3.1.2.1 LIST CACHE

Displays the flows present in the cache. You can specify a text string so that only flows containing this string are shown.

Syntax:

```
NETFLOW Mon+list cache ?
  <cr>
  <word>    Text
```

Example 1:

```
NETFLOW Mon+list cache

Date: 07/01/16 18:46:09

   SrcIf       SrcIPaddress    DstIf       DstIPaddress    AppId    ToS Pr SrcP  DstP  Pkts    Bytes     Exp Life
  ------------------------------------------------------------------------------------------------------------
   ethernet0/0 192.168.212.176 ethernet0/0 216.58.211.238  L4:443   00  6  31235 443   3       121       0   30
        ssl host: "google.com"
   ethernet0/0 216.58.211.238  ethernet0/0 192.168.212.176 L4:443   00  6  443   31235 4       235       0   30
        ssl host: "google.com"
   local       192.168.212.170 ethernet0/0 192.168.212.176 L4:9996  00  17 1027  9996  45      12372     6   54

Printed 3 out of 3

NETFLOW Mon+
```

Example 2:

```
NETFLOW Mon+list cache 192.168.212.170
Date: 07/01/16 18:46:32

   SrcIf       SrcIPaddress    DstIf       DstIPaddress    AppId    ToS Pr SrcP  DstP  Pkts    Bytes     Exp Life
  ------------------------------------------------------------------------------------------------------------
   local       192.168.212.170 ethernet0/0 192.168.212.176 L4:9996  00  17 1027  9996  45      12372     10  20

Printed 1 out of 3
```

```
NETFLOW Mon+
```

3.1.2.2 LIST STATISTICS

Lists the global statistics on the NETFLOW/IPFIX protocol.

Reports the number of active flows, IP packets processed, ignored, etc.

Syntax:

```
NETFLOW Mon+list statistics
NETFLOW Mon+
```

Example:

```
NETFLOW Mon+list statistics

Number of active flows: 56
Packets processed: 472511
Fragments: 0
Ipsec packets: 0
Not sampled packets: 0
Flows expired: 69685 (0 forced)
Flows exported: 69685 in 20584 packets (0 failures)
Option packets: 474
Sampling factor 1 out of 1

NETFLOW Mon+
```

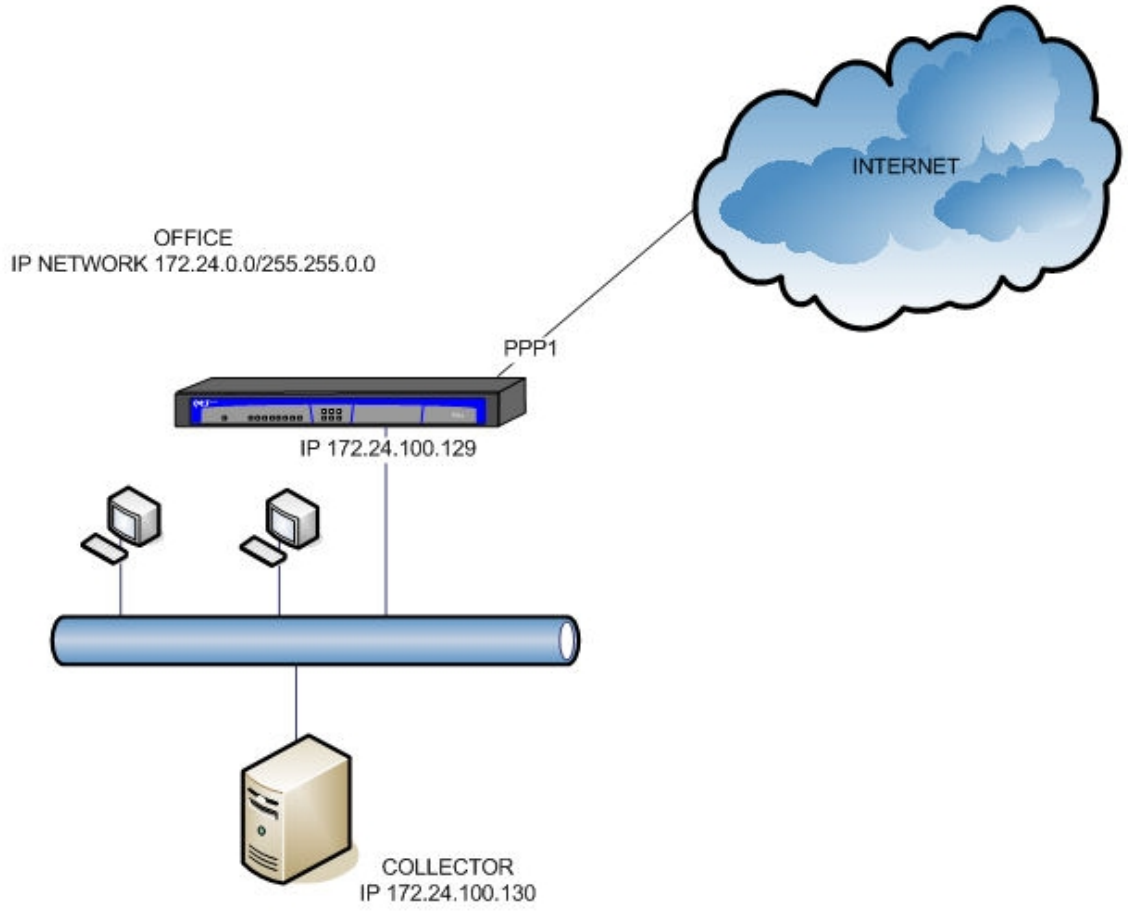
Chapter 4 Examples

4.1 Monitoring IP traffic through netflow

We want to monitor network traffic processed by a router acting as output to Internet. The interface selected for monitoring is the output to Internet, ppp1, and we want to monitor both the input and the output for said interface. We have decided to use netflow protocol version 9 for this.

We have installed collector software in a PC with address 172.24.100.130, using default port 9996. We want the source IP (used by the router to send netflow UDP packets) to be 172.24.100.129.

The following figure represents the scenario described above:



Since we need the highest resolution possible, activity time-out is set to 1 minute.

The netflow protocol is configured as follows:

```
network ppp1
; -- Generic PPP User Configuration --
      ip flow egress
      ip flow ingress
exit
feature netflow
      ip cache timeout active 1
      ip export destination 172.24.100.130
;
      ip export source 172.24.100.129
      ip export version 9
exit
```

4.2 L7 Application information visualization

We want to modify the previous example so that L7 information can be gathered from the traffic using the AFS app-detect feature. The goal is to detect HTTP and SSL flows and extract the hostnames.

AFS must be enabled and both HTTP and SSL detectors configured to detect hostnames. The host part of the Referer Header field provides better information for HTTP because it carries the original host (as entered by the user) that gave rise to a certain HTTP request. If there is no Referer in the HTTP Request, this then falls back to the Host header.

A two-level domain filter is also configured to lower the number of different hosts exported and group more flows under the same hostname (e.g. play.google.com and drive.google.com are both google.com).

The NETFLOW/IPFIX configuration is modified to use the IPFIX protocol, which allows variable size subAppID information elements to be exported. *http-referer* and *ssl-server* are included in the flow record information through the **collect** command.

The application ID is also exported with each flow through **collect app-id**.

Finally, the Interface Table Option Template is configured to be exported. It contains SNMP ifIndex values linked to the interface name and interface description mapping.

The configuration for this is as follows.

```
feature afs
   enable
   app-detect http
   app-detect http referer host-only keep-lvls 2
   app-detect ssl
   app-detect ssl host keep-lvls 2
exit
;
network ppp1
; -- Generic PPP User Configuration --
      ip flow egress
      ip flow ingress
exit
;
feature netflow
      collect app-id
      collect http-referer not-found-txt unknown.ref
      collect ssl-server not-found-txt unknown.ssl
      ip cache timeout active 1
      ip export destination 172.24.100.130
;
      ip export interface-table
      ip export source 172.24.100.129
      ip export version ipfix
exit
```

Chapter 5 Annex A

5.1 Third Party Software

When it comes to TLS negotiation, CIT uses the OpenSSL library code.

Please see a copy of the OpenSSL license below:

The OpenSSL toolkit remains under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. The actual license texts can be found below.

OpenSSL License

Redistribution and use in source and binary forms, with or without modification, are permitted provided the following conditions are met:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project to be used in the OpenSSL Toolkit. (http://www.openssl.org/)"

(4) The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. To obtain written permission, please contact openssl-core@openssl.org.

(5) Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without the OpenSSL Project's prior written permission.

(6) Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project to be used in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USAGE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) IN ANY WAY ARISING FROM THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms, save Tim Hudson (tjh@cryptsoft.com) is the holder.

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the fol-

lowing conditions are met:

(1) Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".
The word 'cryptographic' can be left out if the routines from the library being used are not cryptographically related.

(4) If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) IN ANY WAY ARISING FROM THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license (including the GNU Public License).