Überblick zur IPFIX-Library von Jan

Exportierte Datentypen:

`struct ipfix_exporter`

- enthält alle Exporter-Daten
- der Anwender greift auf die Daten nicht direkt zu, sondern verwendet einen Pointer auf die Datenstruktur als Handler für den Exporter
- bei allen Funktionsaufrufen wird der Pointer auf den betreffenden Exporter mitgegeben

`enum ipfix_transport_protocol {UDP, TCP, SCTP}`

- Datentyp für Transport-Protokolle

Exportierte Funktionen:

Exporter-Management-Funktionen:

`int ipfix_init_exporter (uint32_t source_id, ipfix_exporter** exporter)`

- legt einen neuen Exporter an und initialisiert ihn (Speicher wird von der Bibliothek reserviert)
- Parameter:
  - source_id ist die Exporter-ID
  - über ipfix_exporter wird ein Pointer auf den angelegten Exporter zurückgegeben

`int ipfix_deinit_exporter (ipfix_exporter** exporter)`

- deinitialisiert einen existierenden Exporter und gibt den Speicher frei
- Parameter:
  - exporter ist ein Pointer auf den zu deinitialisierenden Exporter

`int ipfix_add_collector(ipfix_exporter* exporter, char* coll_ip4_addr, int coll_port, ipfix_transport_protocol proto)`

- fügt eine Collector hinzu und öffnet ggf. die entsprechende Verbindung
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - coll_ip4_addr ist die IP-Adresse des Collectors als C-String
  - coll_port ist der Collector-Port
  - protocol bestimmt das zu verwendende Transportprotokoll

`int ipfix_remove_collector(ipfix_exporter* exporter, char* coll_ip4_addr, int coll_port)`

- entfernt eine Collector und schließt ggf. die entsprechende Verbindung
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - coll_ip4_addr ist die IP-Adresse des Collectors als C-String
  - coll_port ist der Collector-Port

Template-Set-Funktionen:

Allgemeines:

- Im Gegensatz zu Daten-Sets legt die Library für jedes Template-Sets eine Buffer an, wo das Template abgelegt werden. Der Speicherplatz wird wieder frei gegeben, wenn das Template gelöscht wird.
- Im Gegensatz zu den Funktionen für die Daten-Sets werden die Template-Parameter in Host-Byte-Order und als Wert (nicht als Referenz) übergeben. Die Library wandelt die Werte bei jedem Aufruf um und speichert sie in einem Buffer.
- Im Gegensatz zu den Daten-Set-Funktionen wird die Gesamtlänge hier von der Bibliothek bestimmt.
- Daten- und Option-Templates unterscheiden sich nur darin, dass sie mit der Funktion ipfix_start_data_template_set oder ipfix_start_option_template_set begonnen werden.

```
int ipfix_start_data_template_set(ipfix_exporter* exporter, uint16_t
      template_id,  uint16_t field_count)
```

- markiert den Beginn für das Anlegen eines neuen Daten-Templates
- Es ist nicht möglich, mehrere Templates gleichzeitig anzulegen, d.h. ipfix_start_data_template_set oder ipfix_start_option_template_set kann kein zweites Mal ausgeführt werden, bevor nicht ipfix_end_template_set das vorhergehende Template abgeschlossen hat.
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - template_id ist die ID
  - field_count ist die Felderanzahl

```
int ipfix_start_options_template_set(ipfix_exporter* exporter, uint16_t
      template_id, uint16_t scope_length, uint16_t option_length)
```

- markiert den Beginn für das Anlegen eines neuen Option-Templates
- Es ist nicht möglich, mehrere Templates gleichzeitig anzulegen, d.h. ipfix_start_data_template_set oder ipfix_start_option_template_set kann kein zweites Mal ausgeführt werden, bevor nicht ipfix_end_template_set das vorhergehende Template abgeschlossen hat.
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - template_id ist die ID
  - scope_length ist die Scope-Länge
  - option_length ist die Option-Länge

```
void ipfix_put_template_field(ipfix_exporter* exporter, uint16_t length,
      uint16_t type, uint32_t enterprise)
```

- hängt ein Feld an das mit ipfix_start_date_template_set bzw. ipfix_start_option_template_set begonnene Template an
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - length ist die Länge des Feldes
  - type ist der Feldtyp
  - enterprise ist das Enterprise-Feld oder Null, wenn das Enterprise-Feld wegfällt

```
int ipfix_end_template_set(ipfix_exporter* exporter)
```

- markiert das Ende eines Templates
- Parameter:
  - exporter ist ein Pointer auf den Exporter

```
int ipfix_remove_template_set(ipfix_exporter* exporter, uint16_t
      template_id)
```

- entfernt ein vom Exporter verwendetes Template und gibt den Speicher frei
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - template_id ist die Template id

Daten-Set-Funktionen

Allgemeines:

- Im Gegensatz zu den Template-Set-Funktionen legt die Library für Daten-Sets keinen Buffer an. Stattdessen werden nur Pointer auf die Speicherbereiche entgegengenommen, die die zu versendenden Daten enthalten. Der Anwender muss also die Daten an den angegebenen Stellen bereithalten, bis sie versendet wurden.
- Die Daten müssen vom Anwender in das richtige Datenformat gebracht werden. Insbesondere übernimmt der Anwender die Umwandlung in Network-Byte-Order (NBO).
- Für Option-Daten-Sets und normale Daten-Sets werden dieselben Funktionen verwendet.

```
int ipfix_start_data_set(ipfix_exporter* exporter, uint16_t* data_length,
      uint16_t* template_id)
```

- markiert den Beginn eines neuen Daten-Sets
- Es ist nicht möglich, mehrere Daten-Sets gleichzeitig anzulegen, d.h. ipfix_start_data_set kann kein zweites Mal ausgeführt werden, bevor nicht ipfix_end_data_set das vorhergehende Daten-Set abgeschlossen hat.
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - data_length ist ein Pointer auf die Gesamtlänge des Daten-Sets in NBO
  - template_id ist ein Pointer auf die ID des verwendeten Templates in NBO

```
void ipfix_put_data_field(ipfix_exporter* exporter, uint16_t length, char*
      data)
```

- hängt ein Feld an das mit ipfix_start_date_set begonnene Daten-Set an
- Parameter:
  - exporter ist ein Pointer auf den Exporter
  - length ist die Länge der zu schreibenden Daten (in host-byte-order!)
  - data ist ein Pointer auf die Daten (ggf. in NBO konvertiert)
- Der mit data angezeigte Datenbereich muss die Daten bereitstellen, bis ipfix_send erfolgreich ausgeführt wurde.

```
int ipfix_end_data_set(ipfix_exporter* exporter)
```

- markiert das Ende eines Daten-Sets
- Parameter:
  - exporter ist ein Pointer auf den Exporter

Sendefunktionen

```
int ipfix_send(ipfix_exporter* exporter)
```

- versendet ein IPFIX-Paket mit den aktuell im Exporter gespeicherten Templates, falls sich die Templates geändert haben oder ein Time-out für das periodische Verschicken der Templates vorliegt
- versendet ein IPFIX-Paket mit den vorher angegebenen Daten-Sets

- Parameter:
    - exporter ist ein Pointer auf den Exporter
- Die mit den Daten-Set-Funktionen angegebenen Speicherbereiche müssen bis zum Aufruf dieser Funktion auf die gültigen Sendedaten zeigen.