RFC 7011 : Specification of the IP Flow Information eXport (IPFIX) Protocol for the Exchange of Flow Information

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 16 septembre 2013

Date de publication du RFC : Septembre 2013

http://www.bortzmeyer.org/7011.html

Le système IPFIX, successeur désigné de Netflow avait été normalisé dans le RFC 5101 [1], que notre RFC met légèrement à jour. Des routeurs de différentes marques peuvent donc désormais envoyer des informations statistiques à des machines d'administration de différentes origines. L'idée de base, héritée de Netflow, est d'**agréger** les informations de trafic, pour que le routeur n'ait à envoyer qu'une synthèse, et pas la totalité du trafic. Comme Netflow, IPFIX peut servir à la gestion du réseau, à la supervision mais aussi à l'espionnage.

Un cahier des charges pour IPFIX avait été spécifié dans le RFC 3917. Le système IPFIX lui-même est normalisé depuis le RFC 5101, ce nouveau RFC prenant en charge le protocole d'autres RFC s'occupant du modèle d'informations (RFC 7012 et RFC 5610) ou bien de l'architecture des différents composants d'IPFIX. C'est ainsi qu'IPFIX a des extensions rigolotes comme la technique de compression du RFC 5473, les données complexes du RFC 6313 ou le stockage de donnéees dans des fichiers permanents du RFC 5655.

La terminologie d'IPFIX est en section 2. Il est recommandé de lire le document d'architecture, le RFC 5470. Un **point d'observation** ("*Observation Point*") est l'endroit du réseau d'où on observe (le port d'un routeur, par exemple). Un domaine d'observation ("*Observation Domain*") est l'ensemble des points d'observation dont les observations vont être agrégées en un seul message IPFIX (un routeur, par exemple). Un **flot** ("*flow*") est un ensemble de paquets, passant par le même point d'observation, et ayant des caractéristiques communes. la définition est volontairement très ouverte. Un flot ne correspond donc pas forcément à une connexion TCP. Par exemple, un flot peut être défini par :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5101.txt

— Les en-têtes du paquet : adresses IP de source et de destination, bien sûr, mais aussi les ports, et/ou des champs applicatifs. Donc, « tous les paquets depuis ou vers `[2001:db8:1:2]:53` » est un flot, identifiant le trafic DNS de `2001:db8:1:2`.
— Les caractéristiques du traitement du paquet : prochain routeur ou interface de sortie.

Un enregistrement de flot ("*flow record*") est composé de la définition du flot et des valeurs mesurées (nombre total de paquets appartenant à ce flot qui ont été vus, nombre total d'octets). Un **gabarit** ("*template*") est une structure de données qui décrit ce qui est contenu dans les enregistrements. Un **exporteur** ("*exporter*"), typiquement le routeur, va émettre ces enregistrements, contenant le résultat de ses observations, vers un **collecteur** ("*collector*") où on fabriquera de jolis graphes.

Notre RFC normalise ensuite le format des paquets (section 3). Comme avec d'autres protocoles, les paquets commencent par un numéro de version, 10 ici (inchangé depuis le RFC 5101), IPFIX marquant sa descendance depuis Netflow, dont la dernière version était la 9. Le RFC décrit aussi l'encodage des données exportées par l'observateur IPFIX (section 6).

Des exemples d'encodage sont fournis dans l'annexe A. Par exemple, A.2.1 utilise des éléments enregistrés à l'IANA <https://www.iana.org/assignments/ipfix/ipfix.xhtml#ipfix-information-elem (IPFIX permet aussi de définir les siens) :

— Adresse IP source `sourceIPv4Address`,
— Adresse IP destination `destinationIPv4Address`,
— Routeur suivant `ipNextHopIPv4Address`,
— Nombre de paquets `packetDeltaCount`,
— Nombre d'octets `octetDeltaCount`.

Les trois premiers identifient le flot et les deux derniers sont les mesures par flot. Le gabarit va donc lister ces cinq élements, avec leur taille (quatre octets chacun).

Enfin, notre RFC décrit le mécanisme de transport, au dessus d'UDP, le protocole traditionnel de Netflow, mais aussi de TCP ou bien du protocole recommandé, SCTP. Pour répondre au cahier des charges, qui insistait sur la sécurité, la section 11.1 décrit aussi l'utilisation de TLS. À propos de sécurité, la section 11 se penche aussi sur la confidentialité (IPFIX permet de transporter des informations sur les métadonnées d'une communication, permettant l'analyse de trafic ; même si on n'a pas le contenu des communications, cela peut être très indiscret). Elle attire aussi l'attention du programmeur sur l'importance de s'assurer de l'intégrité des données. IPFIX pouvant être utilisé pour la facturation, un client malhonnête peut avoir une bonne motivation pour modifier ces données.

La section 1.1 décrit les changements depuis le premier RFC, le RFC 5101. Ils sont de peu d'importance et deux logiciels suivant les deux RFC pourront interagir. Il y a eu plusieurs corrections d'erreurs dans la spécification <http://www.rfc-editor.org/errata_search.php?rfc=5101&presentation=table>, les définitions d'élements d'information situées dans l'ancien RFC ont toute été sorties, vers le registre IANA <https://www.iana.org/assignments/ipfix/ipfix.xml>, qui fait désormais seul autorité. Une nouvelle section, la 5.2, traite le problème du débordement de certains types temporels, débordement qui se produira en 2032 ou 2038 selon les cas (cela semble loin dans le futur mais les logiciels durent longtemps, rappelez-vous le bogue de l'an 2000).

IPFIX est déjà mis en œuvre dans plusieurs systèmes. Un exemple de mise en œuvre en logiciel libre est Maji <http://research.wand.net.nz/software/maji.php>. Mais, apparemment Cacti ne sait pas exporter de l'IPFIX, juste du vieux Netflow. Ntop sait faire mais il faut passer à nProbe l'option `-V 10` (plein d'exemple dans cet article <http://www.plixer.com/blog/scrutinizer/recommended-nprob>, merci à Andrew Feren pour les informations). Si vous voulez voir des traces IPFIX, il y en a deux sur pcapr <http://www.pcapr.net/browse?q=ipfix>. Wireshark sait analyser de l'IPFIX <https://puck.nether.net/pipermail/juniper-nsp/2012-March/022791.html>, il faut lui demander de décoder comme <https://puck.nether.net/pipermail/juniper-nsp/2012-March/022791.html> du cflow <http://www.wireshark.org/docs/dfref/c/cflow.html>. Le RFC

5153 donne des conseils aux programmeurs qui veulent créer ou lire de l'IPFIX. Les RFC 5471 et RFC 5472 sont utiles aux administrateurs réseaux qui voudraient déployer IPFIX. Toutefois, aujourd'hui, des années après la sortie du premier RFC, il semble que peu d'opérateurs utilisent IPFIX (v10). La plupart sont restés bloqués sur Netflow (v9). Les « grands » routeurs (Juniper ou Cisco) savent faire de l'IPFIX mais pas forcément les routeurs de bas de gamme.

Deux autres logiciels importants, des collecteurs IPFIX. Il y a bien sûr pmacct <http://www.pmacct.net/> mais aussi Vflow <https://github.com/VerizonDigital/vflow>. Voici un exemple de fichier de configuration pmacct :

```
# Les bases de données dans lesquelles on stockera
plugins: memory,sqlite3,pgsql

# La version du schéma de données (7 est la plus récente)
sql_table_version: 7

# Le port UDP où les données IPFIX sont envoyées (cela doit coïncider
# avec la config' du routeur)
nfacctd_port: 2055

# Le ou les critères d'agrégation
aggregate: src_host
```

Avec une telle configuration, voici ce qui sera enregistré dans la base PostgreSQL :

```
pmacct=> select ip_src,bytes,packets from acct_v7 ;
    ip_src    | bytes  | packets
--------------+--------+---------
 10.10.86.133 | 371740 |    2083
 192.0.2.2    | 368576 |    2804
(2 lignes)
```

Mais, au lieu de PostgreSQL, on aurait pu aussi utiliser le client en ligne de commandes (grâce au plugin memory) :

```
% pmacct -s
SRC_IP                                  PACKETS              BYTES
192.0.2.2                               2363                 310176
10.10.86.133                            1749                 312036

For a total of: 2 entries
```

Maintenant, si on ajoute au fichier de configuration une agrégation sur d'autres critères :

```
aggregate: src_host,dst_host,proto,src_port,dst_port
```

On aura alors bien plus de lignes dans les bases (une par tuple, et le port source du trafic SSH varie à chaque fois) :

```
% pmacct -s
SRC_IP                                          DST_IP                                          SRC_PORT  DST_
10.10.86.133                                    192.0.2.2                                       57230     22
10.10.86.133                                    192.0.2.2                                       57250     22
...
192.0.2.2                                       192.0.2.1                                       39802     53
192.0.2.2                                       10.10.86.133                                    22        5720

For a total of: 180 entries
```

Et idem dans la base PostgreSQL :

```
pmacct=> select count(*) from acct_v7 ;
 count
-------
   198
(1 ligne)
```

On peut utiliser SQL pour faire des sélections, des agrégations... Exemple, tout le trafic depuis une machine, avec la fonction d'agrégation SQL sum et la fonction de sélection SQL where :

```
pmacct=> select sum(bytes), sum(packets) from acct_v7 where ip_src='192.0.2.2';
  sum   | sum
--------+------
 326807 | 2492
(1 ligne)
```

Ou bien tout le trafic UDP :

```
pmacct=> select sum(bytes), sum(packets) from acct_v7 where ip_proto=17;
  sum   | sum
--------+------
 111738 | 1574
(1 ligne)
```

Par défaut, les données sont agrégées, quel que soit leur âge. Si on veut faire des études historiques, il faut demander à garder les données plus anciennes :

```
aggregate: src_host,dst_host
sql_history: 5m
sql_history_roundoff: m
```

On voit alors les données « tourner » (notez le `order SQL` pour classer par ordre rétro-chronologique) :

```
pmacct=> select ip_src,ip_dst,packets,stamp_inserted,stamp_updated from acct_v7 where ip_dst!='0.0.0.0' orde
    ip_src    |    ip_dst     | packets |   stamp_inserted    |    stamp_updated
--------------+---------------+---------+---------------------+---------------------
 10.10.86.133 | 192.0.2.2     |    1249 | 2016-05-03 10:25:00 | 2016-05-03 10:28:02
 192.0.2.2    | 10.10.86.133  |    1487 | 2016-05-03 10:25:00 | 2016-05-03 10:28:02
 192.0.2.2    | 192.0.2.1     |     116 | 2016-05-03 10:25:00 | 2016-05-03 10:28:02
 192.0.2.2    | 192.0.2.1     |     252 | 2016-05-03 10:20:00 | 2016-05-03 10:26:02
 192.0.2.2    | 10.10.86.133  |    2906 | 2016-05-03 10:20:00 | 2016-05-03 10:26:02
 10.10.86.133 | 192.0.2.2     |    2354 | 2016-05-03 10:20:00 | 2016-05-03 10:26:02
 10.10.86.133 | 192.0.2.2     |    2364 | 2016-05-03 10:15:00 | 2016-05-03 10:21:02
 192.0.2.2    | 192.0.2.1     |     270 | 2016-05-03 10:15:00 | 2016-05-03 10:21:02
 192.0.2.2    | 10.10.86.133  |    2911 | 2016-05-03 10:15:00 | 2016-05-03 10:21:02
 192.0.2.2    | 10.10.86.133  |    1645 | 2016-05-03 10:10:00 | 2016-05-03 10:16:02
 192.0.2.2    | 192.0.2.1     |     154 | 2016-05-03 10:10:00 | 2016-05-03 10:16:02
 10.10.86.133 | 192.0.2.2     |    1310 | 2016-05-03 10:10:00 | 2016-05-03 10:16:02
(12 lignes)
```

(Chaque enregistrement dure un peu plus que les cinq minutes configurées car pmacct « *"buffe-rise"* ».)

Voici enfin un exemple de paquet vu par Wireshark. C'est du vrai IPFIX (numéro de version 10).

```
[Premier paquet, avec les gabarits]

Cisco NetFlow/IPFIX
    Version: 10
    Length: 876
    Timestamp: Jul  8, 2011 21:18:53.000000000 CEST
        ExportTime: 1310152733
    FlowSequence: 0
    Observation Domain Id: 0
    Set 1
        FlowSet Id: Data Template (V10 [IPFIX]) (2)
        FlowSet Length: 248
        Template (Id = 47104, Count = 25)
            Template Id: 47104
            Field Count: 25
            Field (1/25): flowStartMilliseconds
                0... .... .... .... = Pen provided: No
                .000 0000 1001 1000 = Type: flowStartMilliseconds (152)
                Length: 8
            Field (2/25): flowEndMilliseconds
                0... .... .... .... = Pen provided: No
                .000 0000 1001 1001 = Type: flowEndMilliseconds (153)
                Length: 8
            Field (3/25): BYTES_TOTAL
                0... .... .... .... = Pen provided: No
                .000 0000 0101 0101 = Type: BYTES_TOTAL (85)
                Length: 8
            Field (4/25): BYTES_TOTAL [Reverse]
                1... .... .... .... = Pen provided: Yes
                .000 0000 0101 0101 = Type: BYTES_TOTAL (85) [Reverse]
                Length: 8
                PEN: IPFIX Reverse Information Element Private Enterprise (29305)
            Field (5/25): PACKETS_TOTAL
                0... .... .... .... = Pen provided: No
                .000 0000 0101 0110 = Type: PACKETS_TOTAL (86)
                Length: 8
            Field (6/25): PACKETS_TOTAL [Reverse]
                1... .... .... .... = Pen provided: Yes
                .000 0000 0101 0110 = Type: PACKETS_TOTAL (86) [Reverse]
                Length: 8
                PEN: IPFIX Reverse Information Element Private Enterprise (29305)
            Field (7/25): IPV6_SRC_ADDR
                0... .... .... .... = Pen provided: No
                .000 0000 0001 1011 = Type: IPV6_SRC_ADDR (27)
                Length: 16
            Field (8/25): IPV6_DST_ADDR
                0... .... .... .... = Pen provided: No
                .000 0000 0001 1100 = Type: IPV6_DST_ADDR (28)
                Length: 16
            Field (9/25): IP_SRC_ADDR
                0... .... .... .... = Pen provided: No
                .000 0000 0000 1000 = Type: IP_SRC_ADDR (8)
                Length: 4
            Field (10/25): IP_DST_ADDR
                0... .... .... .... = Pen provided: No
                .000 0000 0000 1100 = Type: IP_DST_ADDR (12)
                Length: 4
            Field (11/25): L4_SRC_PORT
                0... .... .... .... = Pen provided: No
                .000 0000 0000 0111 = Type: L4_SRC_PORT (7)
                Length: 2
```

```
    Field (12/25): L4_DST_PORT
        0... .... .... .... = Pen provided: No
        .000 0000 0000 1011 = Type: L4_DST_PORT (11)
        Length: 2
    Field (13/25): PROTOCOL
        0... .... .... .... = Pen provided: No
        .000 0000 0000 0100 = Type: PROTOCOL (4)
        Length: 1
    Field (14/25): flowEndReason
        0... .... .... .... = Pen provided: No
        .000 0000 1000 1000 = Type: flowEndReason (136)
        Length: 1
    Field (15/25): paddingOctets
        0... .... .... .... = Pen provided: No
        .000 0000 1101 0010 = Type: paddingOctets (210)
        Length: 6
    Field (16/25):  21 [pen: CERT Coordination Center]
        1... .... .... .... = Pen provided: Yes
        .000 0000 0001 0101 = Type: 21 [pen: CERT Coordination Center]
        Length: 4
        PEN: CERT Coordination Center (6871)
    Field (17/25): TCP_SEQ_NUM
        0... .... .... .... = Pen provided: No
        .000 0000 1011 1000 = Type: TCP_SEQ_NUM (184)
        Length: 4
    Field (18/25): TCP_SEQ_NUM [Reverse]
        1... .... .... .... = Pen provided: Yes
        .000 0000 1011 1000 = Type: TCP_SEQ_NUM (184) [Reverse]
        Length: 4
        PEN: IPFIX Reverse Information Element Private Enterprise (29305)
    Field (19/25):  14 [pen: CERT Coordination Center]
        1... .... .... .... = Pen provided: Yes
        .000 0000 0000 1110 = Type: 14 [pen: CERT Coordination Center]
        Length: 1
        PEN: CERT Coordination Center (6871)
    Field (20/25):  15 [pen: CERT Coordination Center]
        1... .... .... .... = Pen provided: Yes
        .000 0000 0000 1111 = Type: 15 [pen: CERT Coordination Center]
        Length: 1
        PEN: CERT Coordination Center (6871)
    Field (21/25): 16398 [pen: CERT Coordination Center]
        1... .... .... .... = Pen provided: Yes
        .100 0000 0000 1110 = Type: 16398 [pen: CERT Coordination Center]
        Length: 1
        PEN: CERT Coordination Center (6871)
    Field (22/25): 16399 [pen: CERT Coordination Center]
        1... .... .... .... = Pen provided: Yes
        .100 0000 0000 1111 = Type: 16399 [pen: CERT Coordination Center]
        Length: 1
        PEN: CERT Coordination Center (6871)
    Field (23/25): SRC_VLAN
        0... .... .... .... = Pen provided: No
        .000 0000 0011 1010 = Type: SRC_VLAN (58)
        Length: 2
    Field (24/25): SRC_VLAN [Reverse]
        1... .... .... .... = Pen provided: Yes
        .000 0000 0011 1010 = Type: SRC_VLAN (58) [Reverse]
        Length: 2
        PEN: IPFIX Reverse Information Element Private Enterprise (29305)
    Field (25/25): Unknown(32767)
        0... .... .... .... = Pen provided: No
        .111 1111 1111 1111 = Type: Unknown (32767)
        Length: 65535 [i.e.: "Variable Length"]
Template (Id = 49155, Count = 3)
    Template Id: 49155
    Field Count: 3
    Field (1/3): TCP_SEQ_NUM
        0... .... .... .... = Pen provided: No
        .000 0000 1011 1000 = Type: TCP_SEQ_NUM (184)
```

```
            Length: 4
        Field (2/3):  14 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .000 0000 0000 1110 = Type: 14 [pen: CERT Coordination Center]
            Length: 1
            PEN: CERT Coordination Center (6871)
        Field (3/3):  15 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .000 0000 0000 1111 = Type: 15 [pen: CERT Coordination Center]
            Length: 1
            PEN: CERT Coordination Center (6871)
    Template (Id = 49171, Count = 6)
        Template Id: 49171
        Field Count: 6
        Field (1/6): TCP_SEQ_NUM
            0... .... .... .... = Pen provided: No
            .000 0000 1011 1000 = Type: TCP_SEQ_NUM (184)
            Length: 4
        Field (2/6):  14 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .000 0000 0000 1110 = Type: 14 [pen: CERT Coordination Center]
            Length: 1
            PEN: CERT Coordination Center (6871)
        Field (3/6):  15 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .000 0000 0000 1111 = Type: 15 [pen: CERT Coordination Center]
            Length: 1
            PEN: CERT Coordination Center (6871)
        Field (4/6): 16398 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .100 0000 0000 1110 = Type: 16398 [pen: CERT Coordination Center]
            Length: 1
            PEN: CERT Coordination Center (6871)
        Field (5/6): 16399 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .100 0000 0000 1111 = Type: 16399 [pen: CERT Coordination Center]
            Length: 1
            PEN: CERT Coordination Center (6871)
        Field (6/6): TCP_SEQ_NUM [Reverse]
            1... .... .... .... = Pen provided: Yes
            .000 0000 1011 1000 = Type: TCP_SEQ_NUM (184) [Reverse]
            Length: 4
            PEN: IPFIX Reverse Information Element Private Enterprise (29305)
    Template (Id = 49176, Count = 2)
        Template Id: 49176
        Field Count: 2
        Field (1/2):  18 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .000 0000 0001 0010 = Type: 18 [pen: CERT Coordination Center]
            Length: 65535 [i.e.: "Variable Length"]
            PEN: CERT Coordination Center (6871)
        Field (2/2): 16402 [pen: CERT Coordination Center]
            1... .... .... .... = Pen provided: Yes
            .100 0000 0001 0010 = Type: 16402 [pen: CERT Coordination Center]
            Length: 65535 [i.e.: "Variable Length"]
            PEN: CERT Coordination Center (6871)
    Template (Id = 49156, Count = 2)
        Template Id: 49156
        Field Count: 2
        Field (1/2): SRC_MAC
            0... .... .... .... = Pen provided: No
            .000 0000 0011 1000 = Type: SRC_MAC (56)
            Length: 6
        Field (2/2): DESTINATION_MAC
            0... .... .... .... = Pen provided: No
            .000 0000 0101 0000 = Type: DESTINATION_MAC (80)
            Length: 6

[Deuxième paquet, avec les données]
```

```
Cisco NetFlow/IPFIX
    Version: 10
    Length: 464
    Timestamp: Jul  8, 2011 21:18:53.000000000 CEST
        ExportTime: 1310152733
    FlowSequence: 0
    Observation Domain Id: 0
    Set 1
        FlowSet Id: (Data) (45840)
        FlowSet Length: 448
        Flow 1
            [Duration: 0.274000000 seconds]
                StartTime: Jun  7, 2011 15:22:38.902000000 CEST
                EndTime: Jun  7, 2011 15:22:39.176000000 CEST
            Permanent Octets: 220
            Permanent Octets: 276 (Reverse Type 85 BYTES_TOTAL)
            Permanent Packets: 4
            Permanent Packets: 4 (Reverse Type 86 PACKETS_TOTAL)
            SrcAddr: 77.250.217.161 (77.250.217.161)
            DstAddr: 192.168.5.219 (192.168.5.219)
            SrcPort: 51413
            DstPort: 52026
            Protocol: 6
            Flow End Reason: End of Flow detected (3)
            Enterprise Private entry: (CERT Coordination Center) Type 21: Value (hex bytes): 00 00 00 00
            Vlan Id: 0
            Vlan Id: 0 (Reverse Type 58 SRC_VLAN)
            [Enterprise Private entry: ((null)) Type 32767: Value (hex bytes): 00 c0 13 00 10 14 8d 68 ed 12
                String_len_short: 255
                String_len_short: 17
        Flow 2
            [Duration: 0.672000000 seconds]
                StartTime: Jun  7, 2011 15:22:40.813000000 CEST
                EndTime: Jun  7, 2011 15:22:41.485000000 CEST
            Permanent Octets: 336
            Permanent Octets: 164 (Reverse Type 85 BYTES_TOTAL)
            Permanent Packets: 5
            Permanent Packets: 3 (Reverse Type 86 PACKETS_TOTAL)
            SrcAddr: 192.168.5.219 (192.168.5.219)
            DstAddr: 84.97.86.239 (84.97.86.239)
            SrcPort: 52035
            DstPort: 43572
            Protocol: 6
            Flow End Reason: End of Flow detected (3)
            Enterprise Private entry: (CERT Coordination Center) Type 21: Value (hex bytes): 00 00 00 dd
            Vlan Id: 0
            Vlan Id: 0 (Reverse Type 58 SRC_VLAN)
            [Enterprise Private entry: ((null)) Type 32767: Value (hex bytes): 00 c0 13 00 10 a3 c6 5c 68 02
                String_len_short: 255
                String_len_short: 17
        Flow 3
            [Duration: 2.046000000 seconds]
                StartTime: Jun  7, 2011 15:22:40.813000000 CEST
                EndTime: Jun  7, 2011 15:22:42.859000000 CEST
            Permanent Octets: 336
            Permanent Octets: 164 (Reverse Type 85 BYTES_TOTAL)
            Permanent Packets: 5
            Permanent Packets: 3 (Reverse Type 86 PACKETS_TOTAL)
            SrcAddr: 192.168.5.219 (192.168.5.219)
            DstAddr: 142.68.133.226 (142.68.133.226)
            SrcPort: 52036
            DstPort: 50006
            Protocol: 6
            Flow End Reason: End of Flow detected (3)
            Enterprise Private entry: (CERT Coordination Center) Type 21: Value (hex bytes): 00 00 03 51
            Vlan Id: 0
            Vlan Id: 0 (Reverse Type 58 SRC_VLAN)
            [Enterprise Private entry: ((null)) Type 32767: Value (hex bytes): 00 c0 13 00 10 54 cc ca 16 02
```

```
            String_len_short: 255
            String_len_short: 17
        Flow 4
            [Duration: 0.162000000 seconds]
                StartTime: Jun  7, 2011 15:22:43.806000000 CEST
                EndTime: Jun  7, 2011 15:22:43.968000000 CEST
            Permanent Octets: 60
            Permanent Octets: 40 (Reverse Type 85 BYTES_TOTAL)
            Permanent Packets: 1
            Permanent Packets: 1 (Reverse Type 86 PACKETS_TOTAL)
            SrcAddr: 192.168.5.219 (192.168.5.219)
            DstAddr: 84.221.224.151 (84.221.224.151)
            SrcPort: 52047
            DstPort: 44809
            Protocol: 6
            Flow End Reason: End of Flow detected (3)
            Enterprise Private entry: (CERT Coordination Center) Type 21: Value (hex bytes): 00 00 00 a2
            Vlan Id: 0
            Vlan Id: 0 (Reverse Type 58 SRC_VLAN)
            [Enterprise Private entry: ((null)) Type 32767: Value (hex bytes): 00 c0 13 00 10 60 40 f2 6b 02 00
                String_len_short: 255
                String_len_short: 17
        Flow 5
            [Duration: 0.335000000 seconds]
                StartTime: Jun  7, 2011 15:22:43.804000000 CEST
                EndTime: Jun  7, 2011 15:22:44.139000000 CEST
            Permanent Octets: 60
            Permanent Octets: 40 (Reverse Type 85 BYTES_TOTAL)
            Permanent Packets: 1
            Permanent Packets: 1 (Reverse Type 86 PACKETS_TOTAL)
            SrcAddr: 192.168.5.219 (192.168.5.219)
            DstAddr: 113.160.54.234 (113.160.54.234)
            SrcPort: 52044
            DstPort: 39621
            Protocol: 6
            Flow End Reason: End of Flow detected (3)
            Enterprise Private entry: (CERT Coordination Center) Type 21: Value (hex bytes): 00 00 01 4f
            Vlan Id: 0
            Vlan Id: 0 (Reverse Type 58 SRC_VLAN)
            [Enterprise Private entry: ((null)) Type 32767: Value (hex bytes): 00 c0 13 00 10 a0 86 d6 62 02 00
                String_len_short: 255
                String_len_short: 17
...
```

Questions lectures, vous pouvez regarder cet exposé à FRnog <http://media.frnog.org/FRnOG_11/FRnOG_11-1.pdf>.