Internet Engineering Task Force (IETF)
Request for Comments: 6727
Category: Standards Track
ISSN: 2070-1721

T. Dietz, Ed.
NEC Europe Ltd.
B. Claise
Cisco Systems, Inc.
J. Quittek
NEC Europe Ltd.

October 2012

Definitions of Managed Objects for Packet Sampling

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes extensions to the IPFIX-SELECTOR-MIB module. For IP Flow Information eXport (IPFIX) implementations that use Packet Sampling (PSAMP) techniques, this memo defines the PSAMP-MIB module containing managed objects for providing information on applied packet selection functions and their parameters.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6727.

Table of Contents

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58,RFC 2580 [RFC2580].

2. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document is a product of the IP Flow Information eXport (IPFIX) Working Group (WG). Work on this document was started in the Packet Sampling (PSAMP) WG and moved to the IPFIX WG when the PSAMP WG was concluded.

Its purpose is to define managed objects for monitoring, PSAMP Devices performing packet selection by Sampling and Filtering as described in [RFC5475].

It is assumed that packet Sampling is performed according to the framework defined in [RFC5474]. In this document, the PSAMP terms that appear capitalized are used as defined in [RFC5475].

Managed objects in the PSAMP MIB module are defined as an extension of the IPFIX-MIB and IPFIX-SELECTOR-MIB modules [RFC6615]. Since the IPFIX MIB module is only for monitoring the same holds true for the PSAMP MIB module defined in this document. The definition of objects is in line with the PSAMP information model [RFC5477].

Section 3 gives an overview of the PSAMP documents, while Section 4 refers to the related IPFIX documents. Section 5 describes the structure of the PSAMP MIB module, and Section 6 contains the formal definition. Security issues are discussed in Section 7.

3. Overview of PSAMP Documents

[RFC5474]: "A Framework for Packet Selection and Reporting" describes the PSAMP framework for network elements to select subsets of packets by statistical and other methods, and to export a stream of reports on the selected packets to a Collector.

[RFC5475]: "Sampling and Filtering Techniques for IP Packet Selection" describes the set of packet selection techniques supported by PSAMP.

[RFC5476]: "Packet Sampling (PSAMP) Protocol Specifications" specifies the export of packet information from a PSAMP Exporting Process to a PSAMP Collecting Process.

[RFC5477]: "Information Model for Packet Sampling Exports" defines an information and data model for PSAMP.

This document: "Definitions of Managed Objects for Packet Sampling" describes the PSAMP Management Information Base.

4. Related IPFIX Documents

The IPFIX protocol provides network administrators with access to IP Flow information.

[RFC5101]: "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information" specifies how IPFIX Data Records and Templates are carried via a congestion-aware transport protocol from IPFIX Exporting Processes to IPFIX Collecting Processes. It also specifies the data types used in the PSAMP MIB module and their encoding.

[RFC6615]: The IPFIX-MIB "Definitions of Managed Objects for IP Flow Information Export" is the basis for this document because it extends the IPFIX SELECTOR MIB module defined there.

5. Structure of the PSAMP MIB module

The IPFIX-MIB module defined in [RFC6615] has the concept of a packet Selection Process containing a set of Selector function instances. Selection Processes and functions are referenced in the ipfixSelectionProcessTable of the IPFIX-MIB module. The ipfixSelectionProcessTable identifies an instance of a Selector function by an OID. The OID points to an object that describes the Selector function. For simple Selector functions without parameters, the OID refers to an object that contains only one additional object indicating the current availability of the function. For functions

that have one or more parameters, the object has a subtree that, in addition to an availability object, contains a table with a conceptual column for each parameter. Entries (conceptual rows) in this table represent different combinations of parameter values for instances of the Selector function.

The object ipfixSelectorFunctions in the IPFIX SELECTOR MIB module serves as the root for objects that describe instances of packet Selector functions. The IPFIX SELECTOR MIB module is a very small module that is defined in [RFC6615]. The top-level OIDs of the parameter trees located beneath ipfixSelectorFunctions are maintained by IANA. In the IPFIX SELECTOR MIB module as defined by [RFC6615], the object ipfixSelectorFunctions contains just a single trivial packet Selector function called ipfixFuncSelectAll that selects every packet and has no parameter:

```
ipfixSelectorMIB
+- ipfixSelectorObjects(1)
   +- ipfixSelectorFunctions(1)
      +- ipfixFuncSelectAll(1)
         +- ipfixFuncSelectAllAvail(1)
```

The PSAMP MIB module defined in this document registers additional top-level OIDs for the parameter subtrees of its Selector functions in the IPFIX-SELECTOR-MIB Function sub-registry according to the procedures defined in [RFC6615]. It introduces six new subtrees beneath ipfixSelectorFunctions. Each of them describes a packet Selector function with one or more parameters. Naming and ordering of objects is fully in line with the guidelines given in Section 6.1 of [RFC6615]. All functions and their parameters are already listed in the overview of functions given by the table in Section 8.2.1 of [RFC5477].

5.1. Textual Conventions

The PSAMP MIB module imports two textual conventions that define data types used in this MIB module from other MIB modules. The Unsigned64TC data type is imported from the APPLICATION MIB module [RFC2564], and the Float64TC data type is imported from the FLOAT-TC-MIB module [RFC6340]. Those data types are defined according to [RFC5101]. Those data types are not an integral part of [RFC2578] but are needed to define objects in this MIB module that conform to the Information Elements defined for those objects in [RFC5477].

The Unsigned64TC textual convention describes an unsigned integer of 64 bits. It is imported from the APPLICATION MIB module. The Float64TC textual convention describes the format that is used for 64-bit floating point numbers.

5.2. Packet Selection Functions

In general, different packet Selector functions have different parameters. The PSAMP MIB module contains six objects with subtrees that provide information on parameters of function instances of different Selector functions. All objects are named and structured according to Section 8.2.1 of [RFC5477]:

```
ipfixSelectorFunctions(1)
+-- psampSampCountBased(2)
+-- psampSampTimeBased(3)
+-- psampSampRandOutOfN(4)
+-- psampSampUniProb(5)
+-- psampFiltPropMatch(6)
+-- psampFiltHash(7)
```

Indexing of these functions in the PSAMP MIB module starts with index (2). The function ipfixFuncSelectAll with index (1) is already defined in the IPFIX SELECTOR MIB module as shown above.

The object tree for each of these functions is described below. Semantics of all functions and their parameters are described in detail in [RFC5475]. More information on the Selector Reports can also be found in Section 6.5.2 of [RFC5476].

5.2.1. Systematic Count-Based Sampling

The first Selector function is systematic count-based Sampling. Its availability is indicated by object psampSampCountBasedAvail. The function has two parameters: psampSampCountBasedInterval and psampSampCountBasedSpace. Different combinations of values of these parameters for different instances of the Selector function are represented by different conceptual rows in the table psampSampCountBasedParamSetTable:

```
psampSampCountBased(2)
+-- psampSampCountBasedAvail(1)
+-- psampSampCountBasedParamSetTable(2)
   +-- psampSampCountBasedParamSetEntry(1) [psampSampCountBasedIndex]
      +-- psampSampCountBasedIndex(1)
      +-- psampSampCountBasedInterval(2)
      +-- psampSampCountBasedSpace(3)
```

5.2.2. Systematic Time-Based Sampling

The second Selector function is systematic time-based Sampling. The structure of the subtree for this function is similar to the psampSampCountBased subtree. Parameters are

psampSampTimeBasedInterval and psampSampTimeBasedSpace. They appear to be the same as for count-based Sampling, but their data types are different because they indicate time values instead of numbers of packets:

```
psampSampTimeBased(3)
+-- psampSampTimeBasedAvail(1)
+-- psampSampTimeBasedParamSetTable(2)
   +-- psampSampTimeBasedParamSetEntry(1) [psampSampTimeBasedIndex]
      +-- psampSampTimeBasedIndex(1)
      +-- psampSampTimeBasedInterval(2)
      +-- psampSampTimeBasedSpace(3)
```

5.2.3. Random n-out-of-N Sampling

The third Selector function is random n-out-of-N Sampling. Parameters are psampSampRandOutOfNSize and psampSampRandOutOfNPopulation:

```
psampSampRandOutOfN(4)
+-- psampSampRandOutOfNAvail(1)
+-- psampSampRandOutOfNParamSetTable(2)
   +-- psampSampRandOutOfNParamSetEntry(1) [psampSampRandOutOfNIndex]
      +-- psampSampRandOutOfNIndex(1)
      +-- psampSampRandOutOfNSize(2)
      +-- psampSampRandOutOfNPopulation(3)
```

5.2.4. Uniform Probabilistic Sampling

The fourth Selector function is uniform probabilistic Sampling. It has just a single parameter called psampSampUniProbProbability:

```
psampSampUniProb(5)
+-- psampSampUniProbAvail(1)
+-- psampSampUniProbParamSetTable(2)
   +-- psampSampUniProbParamSetEntry(1) [psampSampUniProbIndex]
      +-- psampSampUniProbIndex(1)
      +-- psampSampUniProbProbability(2)
```

5.2.5. Property Match Filtering

The fifth Selector function is property match Filtering. For this Selector function, there is a broad variety of possible parameters that could be used. But, as stated in Section 8.2.1 of [RFC5477], there are no agreed parameters specified and the subtree for this function only contains an object indicating the availability of this function. Parameters cannot be retrieved via the PSAMP MIB module:

```
psampFiltPropMatch(6)
+-- psampFiltPropMatchAvail(1)
```

5.2.6. Hash-Based Filtering

The sixth Selector function is hash-based Filtering. The object psampFiltHashFunction is an enumeration that specifies the kind of hash function that is applied. These hash functions have quite a number of parameters, and the actual number may vary with the choice of the hash function applied. The common parameter set for all hash-based Filtering functions contains 7 parameters: psampFiltHashInitializerValue, psampFiltHashIpPayloadOffset, psampFiltHashIpPayloadSize, psampFiltHashSelectedRangeMin, psampFiltHashSelectedRangeMax, psampFiltHashOutputRangeMin, and psampFiltHashOutputRangeMax.

```
psampFiltHash(7)
+-- psampFiltHashAvail(1)
+-- psampFiltHashCapabilities(2)
+-- psampFiltHashParamSetTable(3)
   +-- psampFiltHashParamSetEntry(1) [psampFiltHashIndex]
      +-- psampFiltHashIndex(1)
      +-- psampFiltHashFunction(2)
      +-- psampFiltHashInitializerValue(3)
      +-- psampFiltHashIpPayloadOffset(4)
      +-- psampFiltHashIpPayloadSize(5)
      +-- psampFiltHashSelectedRangeMin(6)
      +-- psampFiltHashSelectedRangeMax(7)
      +-- psampFiltHashOutputRangeMin(8)
      +-- psampFiltHashOutputRangeMax(9)
```

Further parameters depend on the applied hash function and are not specified within the PSAMP MIB module.

6. Definitions

```
PSAMP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Integer32, Unsigned32, mib-2
        FROM SNMPv2-SMI                  -- RFC 2578
    TruthValue
        FROM SNMPv2-TC                   -- RFC 2579
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF                 -- RFC 2580
    Unsigned64TC
        FROM APPLICATION-MIB             -- RFC 2564
    Float64TC
        FROM FLOAT-TC-MIB                -- RFC 6340
    ipfixSelectorFunctions
        FROM IPFIX-SELECTOR-MIB;         -- RFC 6615

psampMIB MODULE-IDENTITY
    LAST-UPDATED "201209051200Z"         -- 5 September 2012
    ORGANIZATION "IETF IPFIX Working Group"
    CONTACT-INFO
        "WG charter:
          http://datatracker.ietf.org/wg/ipfix/charter/

        Mailing Lists:
          General Discussion: ipfix@ietf.org
          To Subscribe: https://www.ietf.org/mailman/listinfo/ipfix
          Archive:
     http://www.ietf.org/mail-archive/web/ipfix/current/maillist.html

          Thomas Dietz (editor)
          NEC Europe Ltd.
          NEC Laboratories Europe
          Network Research Division
          Kurfuersten-Anlage 36
          69115 Heidelberg
          Germany
          Phone: +49 6221 4342-128
          EMail: Thomas.Dietz@neclab.eu

          Benoit Claise
          Cisco Systems, Inc.
          De Kleetlaan 6a b1
          Diegem 1831
          Belgium
          Phone:  +32 2 704 5622
          EMail: bclaise@cisco.com
```

```
            Juergen Quittek
            NEC Europe Ltd.
            NEC Laboratories Europe
            Network Research Division
            Kurfuersten-Anlage 36
            69115 Heidelberg
            Germany
            Phone: +49 6221 4342-115
            EMail: quittek@neclab.eu"
          DESCRIPTION
          "The PSAMP MIB defines managed objects for packet sampling
          and filtering.

          These objects provide information about managed nodes
          supporting packet sampling, including packet sampling
          capabilities, configuration, and statistics.
          The PSAMP MIB module registers additional top-level OIDs for
          the parameter subtrees of its Selector functions in the
          IPFIX-SELECTOR-MIB Function sub-registry according to the
          procedures defined in RFC 6615.

          Copyright (c) 2012 IETF Trust and the persons identified
          as authors of the code. All rights reserved.

          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (http://trustee.ietf.org/license-info).

          This version of this MIB module is part of RFC 6727; see the
          RFC itself for full legal notices."
       --  Revision history
       REVISION     "201209051200Z"         -- 5 September 2012
       DESCRIPTION
           "Initial version, published as RFC 6727."
      ::= { mib-2 212 }

   -- Top-level structure of the MIB

   psampObjects     OBJECT IDENTIFIER ::= { psampMIB 1 }
   psampConformance OBJECT IDENTIFIER ::= { psampMIB 2 }

   --====================================================================
   -- Packet selection sampling methods group of objects
   --====================================================================
```

```
   --====================================================================
   --* Method 1: Systematic count-based Sampling
   --====================================================================

   -- Reference: RFC 5475 (Section 5.1), RFC 5476 (Section 6.5.2.1),
   --            and RFC 5477 (Section 8.2)
   psampSampCountBased OBJECT IDENTIFIER
       ::= { ipfixSelectorFunctions 2 }

   psampSampCountBasedAvail OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the availability of systematic
           count-based sampling at the managed node.

           A Selector may be unavailable if it is implemented but
           currently disabled due to, e.g., administrative reasons, lack
           of resources, or similar."
       ::= { psampSampCountBased 1 }

   -- Parameter Set Table ++++++++++++++++++++++++++++++++++++++++++++

   psampSampCountBasedParamSetTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF
                   PsampSampCountBasedParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table lists configurations of systematic count-based
           packet sampling.  A parameter set describing a
           configuration contains two parameters: the sampling
           interval length and space."
       ::= { psampSampCountBased 2 }

   psampSampCountBasedParamSetEntry OBJECT-TYPE
       SYNTAX      PsampSampCountBasedParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Defines an entry in the psampSampCountBasedParamSetTable."
       INDEX { psampSampCountBasedIndex }
       ::= { psampSampCountBasedParamSetTable 1 }

   PsampSampCountBasedParamSetEntry ::=
       SEQUENCE {
           psampSampCountBasedIndex     Integer32,
```

```
            psampSampCountBasedInterval  Unsigned32,
            psampSampCountBasedSpace     Unsigned32
        }

    psampSampCountBasedIndex OBJECT-TYPE
        SYNTAX      Integer32 (1..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index of this parameter set in the
            psampSampCountBasedParamSetTable.  It is used in the
            object ipfixSelectionProcessSelectorFunction entries of
            the ipfixSelectionProcessTable in the IPFIX-MIB as reference
            to this parameter set."
        ::= { psampSampCountBasedParamSetEntry 1 }

    psampSampCountBasedInterval OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "packets"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the number of packets that are
            consecutively sampled.  A value of 100 means that 100
            consecutive packets are sampled."
        REFERENCE
            "RFC 5475 (Section 5.1) and RFC 5477 (Section 8.2)"
        ::= { psampSampCountBasedParamSetEntry 2 }

    psampSampCountBasedSpace OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "packets"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the number of packets between two
            intervals specified by the object
            psampSampCountBasedInterval.  A value of 100 means that
            the next interval starts 100 packets (which are not sampled)
            after the current psampSampCountBasedInterval is over."
        REFERENCE
            "RFC 5475 (Section 5.1) and RFC 5477 (Section 8.2)"
        ::= { psampSampCountBasedParamSetEntry 3 }

    --====================================================================
    --* Method 2: Systematic time-based Sampling
    --====================================================================
```

```
   -- Reference: RFC 5475 (Section 5.1), RFC 5476 (Section 6.5.2.2),
   --            and RFC 5477 (Section 8.2)
   psampSampTimeBased OBJECT IDENTIFIER
       ::= { ipfixSelectorFunctions 3 }

   psampSampTimeBasedAvail OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the availability of systematic
           time-based sampling at the managed node.

           A Selector may be unavailable if it is implemented but
           currently disabled due to, e.g., administrative reasons, lack
           of resources, or similar."
       ::= { psampSampTimeBased 1 }

   -- Parameter Set Table ++++++++++++++++++++++++++++++++++++++++++++

   psampSampTimeBasedParamSetTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF
                   PsampSampTimeBasedParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table lists configurations of systematic time-based
           packet sampling.  A parameter set describing a configuration
           contains two parameters: the sampling interval length and
           the space."
       ::= { psampSampTimeBased 2 }

   psampSampTimeBasedParamSetEntry OBJECT-TYPE
       SYNTAX      PsampSampTimeBasedParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Defines an entry in the psampSampTimeBasedParamSetTable."
       INDEX { psampSampTimeBasedIndex }
       ::= { psampSampTimeBasedParamSetTable 1 }

   PsampSampTimeBasedParamSetEntry ::=
       SEQUENCE {
           psampSampTimeBasedIndex     Integer32,
           psampSampTimeBasedInterval  Unsigned32,
           psampSampTimeBasedSpace     Unsigned32
       }
```

```
   psampSampTimeBasedIndex OBJECT-TYPE
       SYNTAX      Integer32 (1..2147483647)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The index of this parameter set in the
           psampSampTimeBasedParamSetTable.  It is used in the
           object ipfixSelectionProcessSelectorFunction entries of
           the ipfixSelectionProcessTable in the IPFIX-MIB as reference
           to this parameter set."
       ::= { psampSampTimeBasedParamSetEntry 1 }

   psampSampTimeBasedInterval OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "microseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object specifies the time interval in microseconds
           during which all arriving packets are sampled."
       REFERENCE
           "RFC 5475 (Section 5.1) and RFC 5477 (Section 8.2)"
       ::= { psampSampTimeBasedParamSetEntry 2 }

   psampSampTimeBasedSpace OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "microseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object specifies the time interval in microseconds
           between two intervals specified by the object
           psampSampTimeBasedInterval.  A value of 100 means that the
           next interval starts 100 microseconds (during which no
           packets are sampled) after the current
           psampSampTimeBasedInterval is over."
       REFERENCE
           "RFC 5475 (Section 5.1) and RFC 5477 (Section 8.2)"
       ::= { psampSampTimeBasedParamSetEntry 3 }

   --============================================================
   --* Method 3: Random n-out-of-N Sampling
   --============================================================

   -- Reference: RFC 5475 (Section 5.2.1), RFC 5476 (Section 6.5.2.3),
   --            and RFC 5477 (Section 8.2)
   psampSampRandOutOfN OBJECT IDENTIFIER
       ::= { ipfixSelectorFunctions 4 }
```

```
   psampSampRandOutOfNAvail OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the availability of random n-out-of-N
           sampling at the managed node.

           A Selector may be unavailable if it is implemented but
           currently disabled due to, e.g., administrative reasons, lack
           of resources, or similar."
       ::= { psampSampRandOutOfN 1 }

   -- Parameter Set Table ++++++++++++++++++++++++++++++++++++++++++++++

   psampSampRandOutOfNParamSetTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF
                   PsampSampRandOutOfNParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table lists configurations of random n-out-of-N
           sampling.  A parameter set describing a configuration
           contains two parameters: the sampling size and the
           parent population."
       ::= { psampSampRandOutOfN 2 }

   psampSampRandOutOfNParamSetEntry OBJECT-TYPE
       SYNTAX      PsampSampRandOutOfNParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Defines an entry in the psampSampRandOutOfNParamSetTable."
       INDEX { psampSampRandOutOfNIndex }
       ::= { psampSampRandOutOfNParamSetTable 1 }

   PsampSampRandOutOfNParamSetEntry ::=
       SEQUENCE {
           psampSampRandOutOfNIndex        Integer32,
           psampSampRandOutOfNSize Unsigned32,
           psampSampRandOutOfNPopulation   Unsigned32
       }

   psampSampRandOutOfNIndex OBJECT-TYPE
       SYNTAX      Integer32 (1..2147483647)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
```

```
            "The index of this parameter set in the
            psampSampRandOutOfNParamSetTable.  It is used in the
            object ipfixSelectionProcessSelectorFunction entries of
            the ipfixSelectionProcessTable in the IPFIX-MIB as reference
            to this parameter set."
        ::= { psampSampRandOutOfNParamSetEntry 1 }

    psampSampRandOutOfNSize OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "packets"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the number of elements taken from the
            parent Population specified in
            psampSampRandOutOfNPopulation."
        REFERENCE
            "RFC 5475 (Section 5.2.1) and RFC 5477 (Section 8.2)"
        ::= { psampSampRandOutOfNParamSetEntry 2 }

    psampSampRandOutOfNPopulation OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "packets"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the number of elements in the parent
            Population."
        REFERENCE
            "RFC 5475 (Section 5.2.1) and RFC 5477 (Section 8.2)"
        ::= { psampSampRandOutOfNParamSetEntry 3 }

    --============================================================
    --* Method 4: Uniform probabilistic Sampling
    --============================================================

    -- Reference: RFC 5475 (Section 5.2.2), RFC 5476 (Section 6.5.2.4),
    --            and RFC 5477 (Section 8.2)
    psampSampUniProb OBJECT IDENTIFIER ::= { ipfixSelectorFunctions 5 }

    psampSampUniProbAvail OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the availability of random uniform
            probabilistic sampling at the managed node.
```

```
            A Selector may be unavailable if it is implemented but
            currently disabled due to, e.g., administrative reasons, lack
            of resources, or similar."
        ::= { psampSampUniProb 1 }

    -- Parameter Set Table ++++++++++++++++++++++++++++++++++++++++++++

    -- Reference: RFC 5475 (Section 5.2.2.1) and RFC 5477 (Section 8.2)
    psampSampUniProbParamSetTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF
                    PsampSampUniProbParamSetEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table lists configurations of random probabilistic
            sampling.  A parameter set describing a configuration
            contains a single parameter only: the sampling probability."
        ::= { psampSampUniProb 2 }

    psampSampUniProbParamSetEntry OBJECT-TYPE
        SYNTAX      PsampSampUniProbParamSetEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Defines an entry in the psampSampUniProbParamSetTable."
        INDEX { psampSampUniProbIndex }
        ::= { psampSampUniProbParamSetTable 1 }

    PsampSampUniProbParamSetEntry ::=
        SEQUENCE {
            psampSampUniProbIndex       Integer32,
            psampSampUniProbProbability Float64TC
        }

    psampSampUniProbIndex OBJECT-TYPE
        SYNTAX      Integer32 (1..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index of this parameter set in the
            psampSampUniProbParamSetTable.  It is used in the
            object ipfixSelectionProcessSelectorFunction entries of
            the ipfixSelectionProcessTable in the IPFIX-MIB as reference
            to this parameter set."
        ::= { psampSampUniProbParamSetEntry 1 }

    psampSampUniProbProbability OBJECT-TYPE
        SYNTAX      Float64TC
```

```
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the probability that a packet is
            sampled, expressed as a value between 0 and 1.  The
            probability is equal for every packet.  A value of 0 means
            no packet is sampled since the probability is 0.  A value
            of 1 means all packets are sampled since the
            probability is 1.  NaN (not a number) and infinity MUST NOT
            be used."
        REFERENCE
            "RFC 5475 (Section 5.2.2.1) and RFC 5477 (Section 8.2)"
        ::= { psampSampUniProbParamSetEntry 2 }

    --=================================================================
    -- Packet selection filtering methods for a group of objects
    --=================================================================

    --=================================================================
    --* Method 5: Property Match filtering
    --=================================================================

    -- Reserves Method 5; see RFC 5475 (Section 6.1), RFC 5476
    -- (Section 6.5.2.5), and RFC 5477
    psampFiltPropMatch OBJECT IDENTIFIER
        ::= { ipfixSelectorFunctions 6 }

    psampFiltPropMatchAvail OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the availability of property match
            filtering at the managed node.

            A Selector may be unavailable if it is implemented but
            currently disabled due to, e.g., administrative reasons, lack
            of resources, or similar."
        ::= { psampFiltPropMatch 1 }

    --=================================================================
    --* Method 6: Hash filtering
    --=================================================================

    -- Reference: RFC 5475 (Section 6.2), RFC 5476 (Section 6.5.2.6),
    --            and RFC 5477 (Section 8.3)
    psampFiltHash OBJECT IDENTIFIER ::= { ipfixSelectorFunctions 7 }
```

```
   psampFiltHashAvail OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the availability of hash filtering
           at the managed node.

           A Selector may be unavailable if it is implemented but
           currently disabled due to, e.g., administrative reasons, lack
           of resources, or similar."
       ::= { psampFiltHash 1 }

   psampFiltHashCapabilities OBJECT IDENTIFIER
       ::= { psampFiltHash 2 }

   -- Parameter Set Table ++++++++++++++++++++++++++++++++++++++++++++++

   -- Reference: RFC 5475, Sections 6.2, 3.8, and 7.1
   psampFiltHashParamSetTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF
                   PsampFiltHashParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table lists configurations of hash filtering.  A
           parameter set describing a configuration contains eight
           parameters describing the hash function."
       ::= { psampFiltHash 3 }

   psampFiltHashParamSetEntry OBJECT-TYPE
       SYNTAX      PsampFiltHashParamSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Defines an entry in the psampFiltHashParamSetTable."
       INDEX { psampFiltHashIndex }
       ::= { psampFiltHashParamSetTable 1 }

   PsampFiltHashParamSetEntry ::=
       SEQUENCE {
           psampFiltHashIndex            Integer32,
           psampFiltHashFunction         INTEGER,
           psampFiltHashInitializerValue Unsigned64TC,
           psampFiltHashIpPayloadOffset  Unsigned64TC,
           psampFiltHashIpPayloadSize    Unsigned64TC,
           psampFiltHashSelectedRangeMin Unsigned64TC,
           psampFiltHashSelectedRangeMax Unsigned64TC,
```

```
            psampFiltHashOutputRangeMin   Unsigned64TC,
            psampFiltHashOutputRangeMax   Unsigned64TC
        }

    psampFiltHashIndex OBJECT-TYPE
        SYNTAX      Integer32 (1..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index of this parameter set in the
            psampFiltHashParamSetTable.  It is used in the
            object ipfixSelectionProcessSelectorFunction entries of
            the ipfixSelectionProcessTable in the IPFIX-MIB as reference
            to this parameter set."
        ::= { psampFiltHashParamSetEntry 1 }

    psampFiltHashFunction OBJECT-TYPE
        SYNTAX      INTEGER {
                        crc32(1),
                        ipsx(2),
                        bob(3)
                    }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The hash function used by this filter.  The PSAMP-MIB
            defines the following hash functions:

            crc32(1): The CRC-32 Hash Function as defined in RFC 1141.

            ipsx(2): The IPSX Hash Function as described in RFC 5475,
                Appendix A.1.

            bob(3): The BOB Hash Function as described in RFC 5475,
                Appendix A.2.
            "
        REFERENCE
            "RFC 5475 (Section 6.2 and Appendixes A.1 and A.2)
             and RFC 1141"
        ::= { psampFiltHashParamSetEntry 2 }

    psampFiltHashInitializerValue OBJECT-TYPE
        SYNTAX      Unsigned64TC
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the initializer value to the hash
            function."
```

```
        REFERENCE
            "RFC 5475, Sections 6.2, 3.8, and 7.1"
        ::= { psampFiltHashParamSetEntry 3 }

    psampFiltHashIpPayloadOffset OBJECT-TYPE
        SYNTAX      Unsigned64TC
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the IP payload offset used by a
            Hash-based Selection Selector."
        REFERENCE
            "RFC 5475, Sections 6.2, 3.8, and 7.1"
        ::= { psampFiltHashParamSetEntry 4 }

    psampFiltHashIpPayloadSize OBJECT-TYPE
        SYNTAX      Unsigned64TC
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the IP payload size used by a
            Hash-based Selection Selector."
        REFERENCE
            "RFC 5475, Sections 6.2, 3.8, and 7.1"
        ::= { psampFiltHashParamSetEntry 5 }

    psampFiltHashSelectedRangeMin OBJECT-TYPE
        SYNTAX      Unsigned64TC
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the value for the beginning of a hash
            function's selected range."
        REFERENCE
            "RFC 5475, Sections 6.2, 3.8, and 7.1"
        ::= { psampFiltHashParamSetEntry 6 }

    psampFiltHashSelectedRangeMax OBJECT-TYPE
        SYNTAX      Unsigned64TC
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object specifies the value for the end of a hash
            function's selected range."
        REFERENCE
            "RFC 5475, Sections 6.2, 3.8, and 7.1"
        ::= { psampFiltHashParamSetEntry 7 }
```

```
   psampFiltHashOutputRangeMin OBJECT-TYPE
       SYNTAX      Unsigned64TC
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object specifies the value for the beginning of a hash
           function's potential output range."
       REFERENCE
           "RFC 5475, Sections 6.2, 3.8, and 7.1"
       ::= { psampFiltHashParamSetEntry 8 }

   psampFiltHashOutputRangeMax OBJECT-TYPE
       SYNTAX      Unsigned64TC
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object specifies the value for the end of a hash
           function's potential output range."
       REFERENCE
           "RFC 5475, Sections 6.2, 3.8, and 7.1"
       ::= { psampFiltHashParamSetEntry 9 }

   --====================================================================
   -- Conformance information
   --====================================================================

   psampCompliances OBJECT IDENTIFIER ::= { psampConformance 1 }
   psampGroups      OBJECT IDENTIFIER ::= { psampConformance 2 }

   --====================================================================
   -- Compliance statements
   --====================================================================

   psampCompliance MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
           "The implementation of all objects is optional and depends
           on the implementation of the corresponding functionality in
           the equipment."
       MODULE  -- this module
           GROUP psampGroupSampCountBased
           DESCRIPTION
               "These objects must be implemented if systematic
               count-based sampling is implemented in the equipment."
           GROUP psampGroupSampTimeBased
           DESCRIPTION
               "These objects must be implemented if systematic
               time-based sampling is implemented in the equipment."
```

```
            GROUP psampGroupSampRandOutOfN
            DESCRIPTION
                "These objects must be implemented if random n-out-of-N
                sampling is implemented in the equipment."
            GROUP psampGroupSampUniProb
            DESCRIPTION
                "These objects must be implemented if uniform
                probabilistic sampling is implemented in the equipment."
            GROUP psampGroupFiltPropMatch
            DESCRIPTION
                "These objects must be implemented if the property match
                filtering is implemented in the equipment."
            GROUP psampGroupFiltHash
            DESCRIPTION
                "These objects must be implemented if hash filtering
                is implemented in the equipment."
        ::= { psampCompliances 1 }

    --=========================================================================
    -- MIB groupings
    --=========================================================================

    psampGroupSampCountBased OBJECT-GROUP
        OBJECTS {
                  psampSampCountBasedAvail,
                  psampSampCountBasedInterval,
                  psampSampCountBasedSpace
                }
        STATUS  current
        DESCRIPTION
            "These objects are needed if count based sampling is
            implemented."
        ::= { psampGroups 1 }

    psampGroupSampTimeBased OBJECT-GROUP
        OBJECTS {
                  psampSampTimeBasedAvail,
                  psampSampTimeBasedInterval,
                  psampSampTimeBasedSpace
                }
        STATUS  current
        DESCRIPTION
            "These objects are needed if time based sampling is
            implemented."
        ::= { psampGroups 2 }

    psampGroupSampRandOutOfN OBJECT-GROUP
        OBJECTS {
```

```
                psampSampRandOutOfNAvail,
                psampSampRandOutOfNSize,
                psampSampRandOutOfNPopulation
              }
        STATUS  current
        DESCRIPTION
            "These objects are needed if random n-out-of-N sampling is
            implemented."
        ::= { psampGroups 3 }

    psampGroupSampUniProb OBJECT-GROUP
        OBJECTS {
                psampSampUniProbAvail,
                psampSampUniProbProbability
              }
        STATUS  current
        DESCRIPTION
            "These objects are needed if uniform probabilistic sampling
            is implemented."
        ::= { psampGroups 4 }

    psampGroupFiltPropMatch OBJECT-GROUP
        OBJECTS {
                psampFiltPropMatchAvail
              }
        STATUS  current
        DESCRIPTION
            "These objects are needed if property match filtering is
            implemented."
        ::= { psampGroups 5 }

    psampGroupFiltHash OBJECT-GROUP
        OBJECTS {
                psampFiltHashAvail,
                psampFiltHashFunction,
                psampFiltHashInitializerValue,
                psampFiltHashIpPayloadOffset,
                psampFiltHashIpPayloadSize,
                psampFiltHashSelectedRangeMin,
                psampFiltHashSelectedRangeMax,
                psampFiltHashOutputRangeMin,
                psampFiltHashOutputRangeMax
              }
        STATUS  current
        DESCRIPTION
            "These objects are needed if hash filtering is implemented."
        ::= { psampGroups 6 }
```

```
END
```

7. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

All tables in this MIB module may be considered sensitive or vulnerable in some network environments because objects in the tables may reveal information about the network infrastructure and device configuration. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) who have legitimate rights to GET or SET (change/create/delete) them.

8. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
psampMIB	{ mib-2 212 }

Further, IANA has registered the following top-level OIDs in the IPFIX-SELECTOR-MIB Functions sub-registry at http://www.iana.org/assignments/smi-numbers according to the procedures set forth in [RFC6615]:

Decimal	Name	Description	Reference
2	psampSampCountBased	Systematic Count-based Sampling	[RFC6727]
3	psampSampTimeBased	Systematic Time-based Sampling	[RFC6727]
4	psampSampRandOutOfN	Random n-out-of-N Sampling	[RFC6727]
5	psampSampUniProb	Universal Probabilistic Sampling	[RFC6727]
6	psampFiltPropMatch	Property Match Filtering	[RFC6727]
7	psampFiltHash	Hash-based Filtering	[RFC6727]

The prerequisites set forth for addition of these OIDs are to be verified based on the content of this document.

9. Acknowledgment

This document is a product of the PSAMP and IPFIX WGs. The authors would like to thank the following persons: Paul Aitken for his detailed review, Dan Romascanu, the MIB doctors, and many more, for the technical reviews and feedback.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2564] Kalbfleisch, C., Krupczak, C., Presuhn, R., and J. Saperia, "Application Management MIB", RFC 2564, May 1999.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.

[RFC6340] Presuhn, R., "Textual Conventions for the Representation of Floating-Point Numbers", RFC 6340, August 2011.

[RFC6615] Dietz, T., Kobayashi, A., Claise, B., and G. Muenz, "Definitions of Managed Objects for IP Flow Information Export", RFC 6615, June 2012.

10.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC5474] Duffield, N., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, March 2009.

[RFC5476] Claise, B., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

Authors' Addresses

Thomas Dietz (editor)
NEC Europe Ltd.
NEC Laboratories Europe
Kurfuersten-Anlage 36
69115 Heidelberg
Germany

Phone: +49 6221 4342-128
EMail: dietz@neclab.eu

Benoit Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
Diegem 1831
Belgium

Phone: +32 2 704 5622
EMail: bclaise@cisco.com

Juergen Quittek
NEC Europe Ltd.
NEC Laboratories Europe
Kurfuersten-Anlage 36
69115 Heidelberg
Germany

Phone: +49 6221 4342-115
EMail: quittek@neclab.eu