Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

D. Fedyk
C. Hopps
LabN Consulting, L.L.C.

February 22, 2021

IP Traffic Flow Security YANG Module
draft-fedyk-ipsecme-yang-iptfs-02

Abstract

This document describes a yang module for the management of IP Traffic Flow Security additions to IKEv2 and IPsec.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a YANG module [RFC7950] for the management of the IP Traffic Flow Security (IP-TFS) extensions as defined in [I-D.ietf-ipsecme-iptfs]. IP-TFS provides enhancements to an IPsec tunnel Security Association to provide improved traffic confidentiality. Traffic confidentiality reduces the ability of traffic analysis to determine identity and correlate observable traffic patterns. IP-TFS offers efficiency when aggregating traffic in fixed size IPsec tunnel packets.

The YANG data model in this document conforms to the Network Management Datastore Architecture defined in [RFC8342].

The only actively published YANG modules for IPsec are found in [I-D.ietf-i2nsf-sdn-ipsec-flow-protection]. This document uses these models as a general IPsec model that can be augmented. The models in [I-D.ietf-i2nsf-sdn-ipsec-flow-protection] provide for an ike and an ikeless model.

1.1. Terminology & Concepts

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview

This document defines configuration and operational parameters of IP traffic flow security (IP-TFS). IP-TFS, defined in [I-D.ietf-ipsecme-iptfs], defines a security association for tunnel mode IPsec with characteristics that improve traffic confidentiality and reduce bandwidth efficiency loss. These documents assume familiarity with IP security concepts described in [RFC4301].

IP-TFS uses tunnel mode to improve confidentiality by hiding inner packet identifiable information, packet size and packet timing. IP-TFS provides a general capability allowing aggregation of multiple packets in uniform size outer tunnel ipsec packets. It maintains the outer packet size by utilizing combinations of aggregating, padding and fragmentating inner packets to fll out the IPsec outer tunnel packet. Zero byte padding is used to fill the packet when no data is available to send.

This document specifies an extensible configuration model for IP-TFS. This version utilizes the capabilities of IP-TFS to configure fixed size IP-TFS Packets that are transmitted at a constant rate. This model is structured to allow for different types of operation through future augmentation.

IP-TFS YANG augments IPsec YANG model from [I-D.ietf-i2nsf-sdn-ipsec-flow-protection]. IP-TFS makes use of IPsec tunnel mode and adds a small number configuration items to tunnel mode IPsec. As defined in [I-D.ietf-ipsecme-iptfs], any SA configured to use IP-TFS supports only IP-TFS packets i.e. no mixed IPsec modes.

The behavior for IP-TFS is controlled by the source. The self-describing format of an IP-TFS packets allows a sending side to adjust the packet-size and timing independently from any receiver. Both directions are also independent, e.g. IP-TFS may be run only in one direction. This means that counters, which are created here for both directions may be 0 or not updated in the case of an SA that uses IP-TFS only in on direction.

Cases where IP-TFS statistics are active for one direction:

- SA one direction - IP-TFS enabled

- SA both directions - IP-TFS only enabled in one direction

Case where IP-TFS statistics are for both directions:

- SA both directions - IP-TFS enable for both directions

The data model uses following constructs for configuration and management:

- Configuration

- Operational State

This YANG module supports configuration of fixed size and fixed rate packets, and elements that may be augmented to support future configuration. The protocol specification [I-D.ietf-ipsecme-iptfs], goes beyond this simple fixed mode of operation by defining a general format for any type of scheme. In this document the outer IPsec packets can be sent with fixed or variable size (without padding). The configuration allows the fixed packet size to be determined by the path MTU. The fixed packet size can also be configured if a value lower than the path MTU is desired.

Other configuration items include:

- Congestion Control. A congestion control setting to allow IP-TFS to reduce the packet rate when congestion is detected.

- Fixed Rate configuration. The IP-TFS tunnel rate can be configured taking into account either layer 2 overhead or layer 3 overhead. Layer 3 overhead is the IP data rate and layer 2 overhead is the rate of bits on the link. The combination of packet size and rate determines the nominal maximum bandwidth and the transmission interval when fixed size packets are used.

- User packet Fragmentation Control. While fragmentation is recommended for improved efficiency, a configuration is provided if users wish to observe the effect no-fragmentation on their data flows.

The YANG operational data allows the readout of the configured parameters as well as the per SA statistics and error counters for IP-TFS. Per SA IPsec packet statistics are provided as a feature and per SA IP-TFS specific statistics as another feature. Both sets of statistics augment the IPsec YANG models with counters that allow observation of IP-TFS packet efficiency.

Draft [I-D.ietf-i2nsf-sdn-ipsec-flow-protection] has a mature set of IPsec YANG management objects.

IP-TFS YANG augments:

- o Yang catalog entry for ietf-i2nsf-ike@2020-10-30.yang

- o Yang catalog entry for ietf-i2nsf-ikeless@20202-10-30.yang

The Security Policy database entry and Security Association entry for an IPsec Tunnel can be augmented with IP-TFS.

3. YANG Management

3.1. YANG Tree

The following is the YANG tree diagram ([RFC8340]) for the IP-TFS extensions.

```
module: ietf-ipsecme-iptfs
  augment /nsfike:ipsec-ike/nsfike:conn-entry/nsfike:spd
            /nsfike:spd-entry/nsfike:ipsec-policy-config
            /nsfike:processing-info/nsfike:ipsec-sa-cfg:
    +--rw traffic-flow-security
       +--rw congestion-control?     boolean
       +--rw packet-size
       |  +--rw use-path-mtu-discovery?   boolean
       |  +--rw outer-packet-size?        uint16
       +--rw (tunnel-rate)?
       |  +--:(l2-fixed-rate)
       |  |  +--rw l2-fixed-rate?    uint64
       |  +--:(l3-fixed-rate)
       |     +--rw l3-fixed-rate?    uint64
       +--rw dont-fragment?          boolean
       +--rw max-aggregation-time?   decimal64
  augment /nsfike:ipsec-ike/nsfike:conn-entry/nsfike:child-sa-info:
    +--ro traffic-flow-security
       +--ro congestion-control?     boolean
       +--ro packet-size
       |  +--ro use-path-mtu-discovery?   boolean
       |  +--ro outer-packet-size?        uint16
       +--ro (tunnel-rate)?
       |  +--:(l2-fixed-rate)
       |  |  +--ro l2-fixed-rate?    uint64
       |  +--:(l3-fixed-rate)
       |     +--ro l3-fixed-rate?    uint64
       +--ro dont-fragment?          boolean
```

```
          +--ro max-aggregation-time?   decimal64
     augment /nsfikels:ipsec-ikeless/nsfikels:spd/nsfikels:spd-entry
               /nsfikels:ipsec-policy-config/nsfikels:processing-info
               /nsfikels:ipsec-sa-cfg:
       +--rw traffic-flow-security
          +--rw congestion-control?     boolean
          +--rw packet-size
          |  +--rw use-path-mtu-discovery?   boolean
          |  +--rw outer-packet-size?        uint16
          +--rw (tunnel-rate)?
          |  +--:(l2-fixed-rate)
          |  |  +--rw l2-fixed-rate?    uint64
          |  +--:(l3-fixed-rate)
          |     +--rw l3-fixed-rate?    uint64
          +--rw dont-fragment?          boolean
          +--rw max-aggregation-time?   decimal64
     augment /nsfikels:ipsec-ikeless/nsfikels:sad/nsfikels:sad-entry:
       +--ro traffic-flow-security
          +--ro congestion-control?     boolean
          +--ro packet-size
          |  +--ro use-path-mtu-discovery?   boolean
          |  +--ro outer-packet-size?        uint16
          +--ro (tunnel-rate)?
          |  +--:(l2-fixed-rate)
          |  |  +--ro l2-fixed-rate?    uint64
          |  +--:(l3-fixed-rate)
          |     +--ro l3-fixed-rate?    uint64
          +--ro dont-fragment?          boolean
          +--ro max-aggregation-time?   decimal64
     augment /nsfike:ipsec-ike/nsfike:conn-entry/nsfike:child-sa-info:
       +--ro ipsec-stats {ipsec-stats}?
       |  +--ro tx-pkts?        uint64
       |  +--ro tx-octets?      uint64
       |  +--ro tx-drop-pkts?   uint64
       |  +--ro rx-pkts?        uint64
       |  +--ro rx-octets?      uint64
       |  +--ro rx-drop-pkts?   uint64
       +--ro iptfs-inner-pkt-stats {iptfs-stats}?
       |  +--ro tx-pkts?              uint64
       |  +--ro tx-octets?            uint64
       |  +--ro rx-pkts?              uint64
       |  +--ro rx-octets?            uint64
       |  +--ro rx-incomplete-pkts?   uint64
       +--ro iptfs-outer-pkt-stats {iptfs-stats}?
          +--ro tx-all-pad-pkts?       uint64
          +--ro tx-all-pad-octets?     uint64
          +--ro tx-extra-pad-pkts?     uint64
          +--ro tx-extra-pad-octets?   uint64
```

```
          +--ro rx-all-pad-pkts?       uint64
          +--ro rx-all-pad-octets?     uint64
          +--ro rx-extra-pad-pkts?     uint64
          +--ro rx-extra-pad-octets?   uint64
          +--ro rx-errored-pkts?       uint64
          +--ro rx-missed-pkts?        uint64
     augment /nsfikels:ipsec-ikeless/nsfikels:sad/nsfikels:sad-entry:
       +--rw ipsec-stats {ipsec-stats}?
       |  +--ro tx-pkts?        uint64
       |  +--ro tx-octets?      uint64
       |  +--ro tx-drop-pkts?   uint64
       |  +--ro rx-pkts?        uint64
       |  +--ro rx-octets?      uint64
       |  +--ro rx-drop-pkts?   uint64
       +--ro iptfs-inner-pkt-stats {iptfs-stats}?
       |  +--ro tx-pkts?              uint64
       |  +--ro tx-octets?            uint64
       |  +--ro rx-pkts?              uint64
       |  +--ro rx-octets?            uint64
       |  +--ro rx-incomplete-pkts?   uint64
       +--ro iptfs-outer-pkt-stats {iptfs-stats}?
          +--ro tx-all-pad-pkts?       uint64
          +--ro tx-all-pad-octets?     uint64
          +--ro tx-extra-pad-pkts?     uint64
          +--ro tx-extra-pad-octets?   uint64
          +--ro rx-all-pad-pkts?       uint64
          +--ro rx-all-pad-octets?     uint64
          +--ro rx-extra-pad-pkts?     uint64
          +--ro rx-extra-pad-octets?   uint64
          +--ro rx-errored-pkts?       uint64
          +--ro rx-missed-pkts?        uint64
```

3.2. YANG Module

The following is the YANG module for managing the IP-TFS extensions.

```
<CODE BEGINS> file "ietf-ipsecme-iptfs@2021-02-22.yang"
module ietf-ipsecme-iptfs {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ipsecme-iptfs";
  prefix iptfs;

  import ietf-i2nsf-ike {
    prefix nsfike;
  }
  import ietf-i2nsf-ikeless {
    prefix nsfikels;
  }
```

```
  organization
    "IETF IPSECME Working Group (IPSECME)";
  contact
    "WG Web:  <https://tools.ietf.org/wg/ipsecme/>
     WG List: <mailto:ipsecme@ietf.org>

     Author: Don Fedyk
             <mailto:dfedyk@labn.net>

     Author: Christian Hopps
             <mailto:chopps@chopps.org>";

  // RFC Ed.: replace XXXX with actual RFC number and
  // remove this note.

  description
    "This module defines the configuration and operational state for
     managing the IP Traffic Flow Security functionality [RFC XXXX].

     Copyright (c) 2020 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
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX
     (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
     full legal notices.";

  revision 2021-02-22 {
    description
      "Initial Revision";
    reference
      "RFC XXXX: IP Traffic Flow Security YANG Module";
  }

  feature ipsec-stats {
    description
      "This feature indicates the device supports
       per SA IPsec statistics";
  }

  feature iptfs-stats {
    description
```

```
      "This feature indicates the device supports
       per SA IP Traffic Flow Security statistics";
  }

  /*--------------------*/
  /*   groupings        */
  /*--------------------*/

  grouping ipsec-tx-stat-grouping {
    description
      "IPsec outbound statistics";
    leaf tx-pkts {
      type uint64;
      config false;
      description
        "Outbound Packet count";
    }
    leaf tx-octets {
      type uint64;
      config false;
      description
        "Outbound Packet bytes";
    }
    leaf tx-drop-pkts {
      type uint64;
      config false;
      description
        "Outbound dropped packets count";
    }
  }

  grouping ipsec-rx-stat-grouping {
    description
      "IPsec inbound statistics";
    leaf rx-pkts {
      type uint64;
      config false;
      description
        "Inbound Packet count";
    }
    leaf rx-octets {
      type uint64;
      config false;
      description
        "Inbound Packet bytes";
    }
    leaf rx-drop-pkts {
      type uint64;
```

```
      config false;
      description
        "Inbound dropped packets count";
    }
  }

  grouping iptfs-inner-tx-stat-grouping {
    description
      "IP-TFS outbound inner packet statistics";
    leaf tx-pkts {
      type uint64;
      config false;
      description
        "Total number of IP-TFS inner packets sent. This
         count is whole packets only.  A fragmented packet
         counts as one packet";
      reference
        "draft-ietf-ipsecme-iptfs";
    }
    leaf tx-octets {
      type uint64;
      config false;
      description
        "Total number of IP-TFS inner octets sent.  This is
         inner packet octets only.  Does not count padding.";
      reference
        "draft-ietf-ipsecme-iptfs";
    }
  }

  grouping iptfs-outer-tx-stat-grouping {
    description
      "IP-TFS outbound inner packet statistics";
    leaf tx-all-pad-pkts {
      type uint64;
      config false;
      description
        "Total number of transmitted IP-TFS packets that
         were all padding with no inner packet data.";
      reference
        "draft-ietf-ipsecme-iptfs section 2.2.3";
    }
    leaf tx-all-pad-octets {
      type uint64;
      config false;
      description
        "Total number transmitted octets of padding added to
         IP-TFS packets with no inner packet data.";
```

```
      reference
        "draft-ietf-ipsecme-iptfs section 2.2.3";
    }
    leaf tx-extra-pad-pkts {
      type uint64;
      config false;
      description
        "Total number of transmitted outer IP-TFS packets
         that included some padding.";
      reference
        "draft-ietf-ipsecme-iptfs section 2.2.3.1";
    }
    leaf tx-extra-pad-octets {
      type uint64;
      config false;
      description
        "Total number of transmitted octets of padding added
         to outer IP-TFS packets with data.";
      reference
        "draft-ietf-ipsecme-iptfs section 2.2.3.1";
    }
  }

  grouping iptfs-inner-rx-stat-grouping {
    description
      "IP-TFS inner packet inbound statistics";
    leaf rx-pkts {
      type uint64;
      config false;
      description
        "Total number of IP-TFS inner packets received.";
      reference
        "draft-ietf-ipsecme-iptfs section 2.2";
    }
    leaf rx-octets {
      type uint64;
      config false;
      description
        "Total number of IP-TFS inner octets received.  Does
         not include padding or overhead";
      reference
        "draft-ietf-ipsecme-iptfs section 2.2";
    }
    leaf rx-incomplete-pkts {
      type uint64;
      config false;
      description
        "Total number of IP-TFS inner packets that were
```

```
           incomplete.  Usually this is due to fragments not
           received. Also, this may be due to misordering or
           errors in received outer packets.";
        reference
          "draft-ietf-ipsecme-iptfs";
      }
    }

    grouping iptfs-outer-rx-stat-grouping {
      description
        "IP-TFS outer packet inbound statistics";
      leaf rx-all-pad-pkts {
        type uint64;
        config false;
        description
          "Total number of received IP-TFS packets that were
           all padding with no inner packet data.";
        reference
          "draft-ietf-ipsecme-iptfs section 2.2.3";
      }
      leaf rx-all-pad-octets {
        type uint64;
        config false;
        description
          "Total number received octets of padding added to
           IP-TFS packets with no inner packet data.";
        reference
          "draft-ietf-ipsecme-iptfs section 2.2.3";
      }
      leaf rx-extra-pad-pkts {
        type uint64;
        config false;
        description
          "Total number of received outer IP-TFS packets that
           included some padding.";
        reference
          "draft-ietf-ipsecme-iptfs section 2.2.3.1";
      }
      leaf rx-extra-pad-octets {
        type uint64;
        config false;
        description
          "Total number of received octets of padding added to
           outer IP-TFS packets with data.";
        reference
          "draft-ietf-ipsecme-iptfs section 2.2.3.1";
      }
      leaf rx-errored-pkts {
```

```
      type uint64;
      config false;
      description
        "Total number of IP-TFS outer packets dropped due to
         errors.";
      reference
        "draft-ietf-ipsecme-iptfs";
    }
    leaf rx-missed-pkts {
      type uint64;
      config false;
      description
        "Total number of IP-TFS outer packets missing
         indicated by missing sequence number.";
      reference
        "draft-ietf-ipsecme-iptfs";
    }
  }

  grouping iptfs-config {
    description
      "This is the grouping for iptfs configuration";
    container traffic-flow-security {
      // config true; want this so we can refine?
      description
        "Configure the IPSec TFS in Security
         Association Database (SAD)";
      leaf congestion-control {
        type boolean;
        default "true";
        description
          "Congestion Control With the congestion controlled
           mode, IP-TFS adapts to network congestion by
           lowering the packet send rate to accommodate the
           congestion, as well as raising the rate when
           congestion subsides.";
        reference
          "draft-ietf-ipsecme-iptfs section 2.5.2";
      }
      container packet-size {
        description
          "Packet size is either auto-discovered or manually
           configured.";
        leaf use-path-mtu-discovery {
          type boolean;
          default "true";
          description
            "Utilize path mtu discovery to determine maximum IP-TFS
```

```
            packet size. If the packet size is explicitly
            configured, then it will only be adjusted downward
            if use-path-mtu-discovery is set.";
         reference
           "draft-ietf-ipsecme-iptfs section 4.2";
       }
       leaf outer-packet-size {
         type uint16;
         description
           "The size of the outer encapsulating tunnel packet (i.e.,
            the IP packet containing the ESP payload).";
         reference
           "draft-ietf-ipsecme-iptfs section 4.2";
       }
     }
     choice tunnel-rate {
       description
         "TFS bit rate may be specified at layer 2 wire
          rate or layer 3 packet rate";
       leaf l2-fixed-rate {
         type uint64;
         description
           "Target bandwidth/bit rate in bps for iptfs tunnel. This
            fixed rate is the nominal timing for the fixed size packet.
            If congestion control is enabled the rate may be adjusted
            down (or up if unset).";
         reference
           "draft-ietf-ipsecme-iptfs section 4.1";
       }
       leaf l3-fixed-rate {
         type uint64;
         description
           "Target bandwidth/bit rate in bps for iptfs tunnel. This
            fixed rate is the nominal timing for the fixed size packet.
            If congestion control is enabled the rate may be adjusted
            down (or up if unset).";
         reference
           "draft-ietf-ipsecme-iptfs section 4.1";
       }
     }
     leaf dont-fragment {
       type boolean;
       default "false";
       description
         "Disable packet fragmentation across consecutive iptfs
          tunnel packets";
       reference
         "draft-ietf-ipsecme-iptfs section 2.2.4 and 6.4.1";
```

```
      }
      leaf max-aggregation-time {
        type decimal64 {
          fraction-digits 6;
        }
        units "milliseconds";
        description
          "Maximum Aggregation Time in  Milliseconds
           or fractional milliseconds down to 1 nanosecond";
      }
    }
  }

  /*
   * IP-TFS ike configuration
   */

  augment "/nsfike:ipsec-ike/nsfike:conn-entry/nsfike:spd/"
        + "nsfike:spd-entry/"
        + "nsfike:ipsec-policy-config/"
        + "nsfike:processing-info/"
        + "nsfike:ipsec-sa-cfg" {
    description
      "IP-TFS configuration for this policy.";
    uses iptfs-config;
  }

  augment "/nsfike:ipsec-ike/nsfike:conn-entry/"
        + "nsfike:child-sa-info" {
    description
      "IP-TFS configured on this SA.";
    uses iptfs-config {
      refine "traffic-flow-security" {
        config false;
      }
    }
  }

  /*
   * IP-TFS ikeless configuration
   */

  augment "/nsfikels:ipsec-ikeless/nsfikels:spd/"
        + "nsfikels:spd-entry/"
        + "nsfikels:ipsec-policy-config/"
        + "nsfikels:processing-info/"
        + "nsfikels:ipsec-sa-cfg" {
    description
```

```
      "IP-TFS configuration for this policy.";
    uses iptfs-config;
  }

  augment "/nsfikels:ipsec-ikeless/nsfikels:sad/"
        + "nsfikels:sad-entry" {
    description
      "IP-TFS configured on this SA.";
    uses iptfs-config {
      refine "traffic-flow-security" {
        config false;
      }
    }
  }

  /*
   * packet counters
   */

  augment "/nsfike:ipsec-ike/nsfike:conn-entry/"
        + "nsfike:child-sa-info" {
    description
      "Per SA Counters";
    container ipsec-stats {
      if-feature "ipsec-stats";
      config false;
      description
        "IPsec per SA packet counters.";
      uses ipsec-tx-stat-grouping {
        //when "direction = 'outbound'";
      }
      uses ipsec-rx-stat-grouping {
        //when "direction = 'inbound'";
      }
    }
    container iptfs-inner-pkt-stats {
      if-feature "iptfs-stats";
      config false;
      description
        "IPTFS per SA inner packet counters.";
      uses iptfs-inner-tx-stat-grouping {
        //when "direction = 'outbound'";
      }
      uses iptfs-inner-rx-stat-grouping {
        //when "direction = 'inbound'";
      }
    }
    container iptfs-outer-pkt-stats {
```

```
      if-feature "iptfs-stats";
      config false;
      description
        "IPTFS per SA outer packets counters.";
      uses iptfs-outer-tx-stat-grouping {
        //when "direction = 'outbound'";
      }
      uses iptfs-outer-rx-stat-grouping {
        //when "direction = 'inbound'";
      }
    }
  }

  /*
   * packet counters
   */

  augment "/nsfikels:ipsec-ikeless/nsfikels:sad/"
        + "nsfikels:sad-entry" {
    description
      "Per SA Counters";
    container ipsec-stats {
      if-feature "ipsec-stats";
      description
        "IPsec per SA packet counters.";
      uses ipsec-tx-stat-grouping {
        //when "direction = 'outbound'";
      }
      uses ipsec-rx-stat-grouping {
        //when "direction = 'inbound'";
      }
    }
    container iptfs-inner-pkt-stats {
      if-feature "iptfs-stats";
      config false;
      description
        "IPTFS per SA inner packet counters.";
      uses iptfs-inner-tx-stat-grouping {
        //when "direction = 'outbound'";
      }
      uses iptfs-inner-rx-stat-grouping {
        //when "direction = 'inbound'";
      }
    }
    container iptfs-outer-pkt-stats {
      if-feature "iptfs-stats";
      config false;
      description
```

```
          "IPTFS per SA outer packets counters.";
        uses iptfs-outer-tx-stat-grouping {
          //when "direction = 'outbound'";
        }
        uses iptfs-outer-rx-stat-grouping {
          //when "direction = 'inbound'";
        }
      }

  }
}
<CODE ENDS>
```

4. IANA Considerations

4.1. Updates to the IETF XML Registry

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registration has been made:

URI:
urn:ietf:params:xml:ns:yang:ietf-ipsecme-iptfs

Registrant Contact:
The IESG.

XML:
N/A; the requested URI is an XML namespace.

4.2. Updates to the YANG Module Names Registry

This document registers one YANG module in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registration has been made:

name:
ietf-ipsecme-iptfs

namespace:
urn:ietf:params:xml:ns:yang:ietf-ipsecme-iptfs

prefix:
iptfs

reference:
RFC XXXX (RFC Ed.: replace XXXX with actual RFC number and remove this note.)

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The YANG module defined in this document can enable, disable and modify the behavior of IP traffic flow security, for the implications regarding these types of changes consult the [I-D.ietf-ipsecme-iptfs] which defines the functionality.

6. Acknowledgements

The authors would like to thank Eric Kinzie for his feedback on the YANG model.

7. References

7.1. Normative References

[I-D.ietf-i2nsf-sdn-ipsec-flow-protection]
Marin-Lopez, R., Lopez-Millan, G., and F. Pereniguez-Garcia, "Software-Defined Networking (SDN)-based IPsec Flow Protection", draft-ietf-i2nsf-sdn-ipsec-flow-protection-12 (work in progress), October 2020.

[I-D.ietf-ipsecme-iptfs]
Hopps, C., "IP-TFS: IP Traffic Flow Security Using Aggregation and Fragmentation", draft-ietf-ipsecme-iptfs-06 (work in progress), January 2021.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

7.2. Informative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Appendix A. Examples

The following examples show configuration and operational data for the ikeless case in xml and ike case in json. Also, the operational statistics for the ikeless case are shown using xml.

A.1. Example XML Configuration

This example illustrates configuration for IP-TFS in the ikeless case. Note that since this augments the ipsec ikeless schema only minimal ikeless configuration to satisfy the schema has been populated.

```
<i:ipsec-ikeless
  xmlns:i="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless"
  xmlns:tfs="urn:ietf:params:xml:ns:yang:ietf-ipsecme-iptfs">
  <i:spd>
    <i:spd-entry>
      <i:name>protect-policy-1</i:name>
      <i:direction>outbound</i:direction>
      <i:ipsec-policy-config>
        <i:traffic-selector>
          <i:local-prefix>1.1.1.1/32</i:local-prefix>
          <i:remote-prefix>2.2.2.2/32</i:remote-prefix>
        </i:traffic-selector>
        <i:processing-info>
          <i:action>protect</i:action>
          <i:ipsec-sa-cfg>
            <tfs:traffic-flow-security>
             <tfs:congestion-control>true</tfs:congestion-control>
              <tfs:packet-size>
                <tfs:use-path-mtu-discovery
                   >true</tfs:use-path-mtu-discovery>
              </tfs:packet-size>
              <tfs:l2-fixed-rate>1000000000</tfs:l2-fixed-rate>
              <tfs:max-aggregation-time
                 >0.1</tfs:max-aggregation-time>
            </tfs:traffic-flow-security>
          </i:ipsec-sa-cfg>
        </i:processing-info>
      </i:ipsec-policy-config>
    </i:spd-entry>
  </i:spd>
</i:ipsec-ikeless>
```


Figure 1: Example IP-TFS XML configuration

A.2. Example XML Operational Data

This example illustrates operational data for IP-TFS in the ikeless case. Note that since this augments the ipsec ikeless schema only minimal ikeless configuration to satisfy the schema has been populated.

```
<i:ipsec-ikeless
  xmlns:i="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless"
  xmlns:tfs="urn:ietf:params:xml:ns:yang:ietf-ipsecme-iptfs">
  <i:sad>
    <i:sad-entry>
      <i:name>sad-1</i:name>
      <i:ipsec-sa-config>
        <i:spi>1</i:spi>
        <i:traffic-selector>
          <i:local-prefix>1.1.1.1/32</i:local-prefix>
          <i:remote-prefix>2.2.2.2/32</i:remote-prefix>
        </i:traffic-selector>
      </i:ipsec-sa-config>
      <tfs:traffic-flow-security>
       <tfs:congestion-control>true</tfs:congestion-control>
        <tfs:packet-size>
          <tfs:use-path-mtu-discovery>true</tfs:use-path-mtu-discovery>
        </tfs:packet-size>
        <tfs:l2-fixed-rate>1000000000</tfs:l2-fixed-rate>
        <tfs:max-aggregation-time>0.100</tfs::max-aggregation-time>
      </tfs:traffic-flow-security>
    </i:sad-entry>
  </i:sad>
</i:ipsec-ikeless>
```


Figure 2: Example IP-TFS XML Operational data

A.3. Example JSON Configuration

This example illustrates config data for IP-TFS in the ike case. Note that since this augments the ipsec ike schema only minimal ike configuration to satisfy the schema has been populated.


```
{
  "ietf-i2nsf-ike:ipsec-ike": {
    "ietf-i2nsf-ike:conn-entry": [
      {
        "name": "my-peer-connection",
        "ike-sa-encr-alg": [
          {
            "id": 1,
            "algorithm-type": 12,
            "key-length": 128
          }
          ],
          "local": {
            "local-pad-entry-name": "local-1"
          },
```

```
            "remote": {
              "remote-pad-entry-name": "remote-1"
            },
            "ietf-i2nsf-ike:spd": {
            "spd-entry": [
              {
                "name": "protect-policy-1",
                "ipsec-policy-config": {
                  "traffic-selector": {
                    "local-prefix": "1.1.1.1/32",
                    "remote-prefix": "2.2.2.2/32"
                  },
                  "processing-info": {
                    "action": "protect",
                    "ipsec-sa-cfg": {
                      "ietf-ipsecme-iptfs:traffic-flow-security": {
                        "congestion-control": "true",
                        "l2-fixed-rate": 1000000000,
                        "packet-size": {
                          "use-path-mtu-discovery": "true"
                        },
                        "max-aggregation-time": "0.1"
                      }
                    }
                  }
                }
              }
            ]
          }
        }
      ]
    }
  }
```


Figure 3: Example IP-TFS JSON configuration

A.4. Example JSON Operational Data

This example illustrates operational data for IP-TFS in the ike case. Note that since this augments the ipsec ike tree only minimal ike configuration to satisfy the schema has been populated.

```
{
  "ietf-i2nsf-ike:ipsec-ike": {
    "ietf-i2nsf-ike:conn-entry": [
      {
        "name": "my-peer-connection",
        "ike-sa-encr-alg": [
        {
          "id": 1,
          "algorithm-type": 12,
          "key-length": 128
        }
        ],
        "local": {
          "local-pad-entry-name": "local-1"
        },
        "remote": {
          "remote-pad-entry-name": "remote-1"
        },
        "ietf-i2nsf-ike:child-sa-info": {
          "ietf-ipsecme-iptfs:traffic-flow-security": {
            "congestion-control": "true",
            "l2-fixed-rate": 1000000000,
            "packet-size": {
              "use-path-mtu-discovery": "true"
            },
            "max-aggregation-time": "0.1"
          }
        }
      }
    ]
  }
}
```


Figure 4: Example IP-TFS JSON Operational data

A.5. Example JSON Operational Statistics

This example shows the json formated statistics for IP-TFS. Note a unidirectional IP-TFS transmit side is illustrated, with arbitray numbers for transmit.


```
{
  "ietf-i2nsf-ikeless:ipsec-ikeless": {
    "sad": {
      "sad-entry": [
        {
          "name": "sad-1",
          "ipsec-sa-config": {
```

```
               "spi": 1,
               "traffic-selector": {
                 "local-prefix": "1.1.1.1/32",
                 "remote-prefix": "2.2.2.2/32"
               }
             },
             "ietf-ipsecme-iptfs:ipsec-stats": {
               "tx-pkts": "300",
               "tx-octets": "80000",
               "tx-drop-pkts": "2",
               "rx-pkts": "0",
               "rx-octets": "0",
               "rx-drop-pkts": "0"
             },
             "ietf-ipsecme-iptfs:iptfs-inner-pkt-stats": {
               "tx-pkts": "250",
               "tx-octets": "75000",
               "rx-pkts": "0",
               "rx-octets": "0",
               "rx-incomplete-pkts": "0"
             },
             "ietf-ipsecme-iptfs:iptfs-outer-pkt-stats": {
               "tx-all-pad-pkts": "40",
               "tx-all-pad-octets": "40000",
               "tx-extra-pad-pkts": "200",
               "tx-extra-pad-octets": "30000",
               "rx-all-pad-pkts": "0",
               "rx-all-pad-octets": "0",
               "rx-extra-pad-pkts": "0",
               "rx-extra-pad-octets": "0",
               "rx-errored-pkts": "0",
               "rx-missed-pkts": "0"
             },
             "ipsec-sa-state": {
               "sa-lifetime-current": {
                 "time": 80000,
                 "bytes": 4000606,
                 "packets": 1000,
                 "idle": 5
               }
             }
           }
         ]
       }
     }
   }
```


Figure 5: Example IP-TFS JSON Statistics

Authors' Addresses

Don Fedyk
LabN Consulting, L.L.C.

Email: dfedyk@labn.net

Christian Hopps
LabN Consulting, L.L.C.

Email: chopps@chopps.org