I2NSF
Internet-Draft
Intended status: Experimental
Expires: May 3, 2017

R. Marin-Lopez
G. Lopez-Millan
University of Murcia
S. Varadhan
Oracle

October 30, 2016

Software-Defined Networking (SDN)-based IPsec Flow Protection
draft-abad-i2nsf-sdn-ipsec-flow-protection-01

Abstract

This document describes the use case of providing IPsec-based flow protection by means of a Software-Defined Network (SDN) controller and raises the requirements to support this service. It considers two main scenarios: (i) gateway-to-gateway and (ii) host-to-gateway (Road Warrior). For the gateway-to-gateway scenario, this document describes a mechanism to support the distribution of IPsec information to flow-based Network Security Functions (NSFs) that implements IPsec to protect data traffic. between network resources to protect data traffic with IPsec and IKE, in intra and inter-SDN cases. The host-to-gateway case defines a mechanism to distribute IPsec information to the NSF to protect data with IPsec between an end user's device (host) and a gateway.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Software-Defined Networking (SDN) is an architecture that enables users to directly program, orchestrate, control and manage network resources through software. SDN paradigm relocates the control of network resources to a dedicated network element, namely SDN controller. The SDN controller manages and configures the distributed network resources and provides an abstracted view of the network resources to the SDN applications. The SDN application can customize and automate the operations (including management) of the abstracted network resources in a programmable manner via this interface [RFC7149][ITU-T.Y.3300] [ONF-SDN-Architecture][ONF-OpenFlow].

Typically, traditional IPsec VPN concentrators and, in general, gateways supporting IKE/IPsec, are configured manually. This makes the IPsec security association (SA) management difficult and generates a lack of flexibility, specially if the number of security policies and SAs to handle is high. With the growth of SDN-based scenarios where network resources are deployed in an autonomous manner, a mechanism to manage IPsec SAs according to the SDN architecture becomes more relevant. Thus, the SDN-based service described in this document will autonomously deal with IPsec-based data protection also in such as an autonomous manner.

IPsec architecture [RFC4301] defines a clear separation between the processing to provide security services to IP packets and the key management procedures to establish the IPsec security association. In this document, we define a service where the key management procedures can be carried by an external entity: the security controller.

First, this document exposes the requirements to support the protection of data flows using IPsec [RFC4301]. We consider two cases:

1) The network resource (or Network Security Function, NSF) implements the Internet Key Exchange (IKE) protocol and the IPsec databases: the Security Policy Database (SPD), the Security Association Database (SAD) and the Peer Authorization Database (PAD). The controller is in charge of provisioning the NSF with the required information about IKE, the SPD and the PAD.

2) The NSF only implements the IPsec databases (no IKE implementation). The controller will provide the required parameters to create valid entries in the PAD, the SPD and the SAD in the NSF. Therefore, the NSF will have only support for IPsec while automated key management functionality is moved to the controller.

In both cases, an interface/protocol will be required to carry out this provisioning between the security controller and the NSF. In particular, it is required the provision of SPD and PAD entries and the credentials and information related with the IKE negotiation (case 1); or the required SPD, PAD and SAD entries with information such as keys, cryptographic algorithms, IP addresses, IPsec protocol (AH or ESP), IPsec protocol mode (tunnel or transport), lifetime of the SA, etc (case 2). An example for case 1 using NETFCONF/YANG can be found in [netconf-vpn]. A YANG model for IPsec can be found in [I-D.tran-ipsecme-yang].

Second, this document considers two scenarios to manage autonomously IPsec SAs: gateway-to-gateway and host-to-gateway [RFC6071]. The gateway-to-gateway scenario shows how flow protection services are useful when data is to be protected across gateways in the network. Each gateway will implement a flow-based NSF. The use case described in Section 10.1 depicts how these services could be used to protect IP traffic among various geographically distributed networks under the domain of the same security controller. A variant of this scenario is also covered in Section 10.2, where the NSFs involved are under the control of different security controllers.

The host-to-gateway scenario described in Section 10.3 covers the case where one end user belonging to a network wants to access securely its network from another external network. In such a case, an IPsec SA needs to be established between the end user's host and the gateway, which is a flow-based NSF. In this document, we describe how the security controller can still configure automatically the IPsec SA in the NSF.

It is worth noting that this work pays attention to the challenge "Lack of Mechanism for Dynamic Key Distribution to NSFs" defined in [I-D.ietf-i2nsf-problem-and-use-cases] in the particular case of the establishment and management of IPsec security associations. In fact, this I-D could be considered as a proper use case for this challenge in [I-D.ietf-i2nsf-problem-and-use-cases].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. When these words appear in lower case, they have their natural language meaning.

3. Terminology

This document uses the terminology described in [RFC7149], [RFC4301], [ITU-T.Y.3300], [ONF-SDN-Architecture], [ONF-OpenFlow], [ITU-T.X.1252], [ITU-T.X.800] and [I-D.ietf-i2nsf-terminology]. In addition, the following terms are defined below:

- Software-Defined Networking. A set of techniques enabling to directly program, orchestrate, control, and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner [ITU-T.Y.3300].

- Flow/Data Flow. Set of network packets sharing a set of characteristics, for example IP dst/src values or QoS parameters.

- Flow Protection Policy. The set of rules defining the conditions under which a data flow MUST be protected with IPsec, and the rules that MUST be applied to the specific flow.

- IKE. Protocol to establish IPsec Security Associations (SAs). It requires information about the required authentication method (i.e. preshared keys), DH groups, modes and algorithms for IKE phase 1, etc.

- SPD. IPsec Security Policy Database. It includes information about IPsec policies direction (in, out), local and remote addresses, inbound and outboud SAs, etc.

- SAD. IPsec Security Associations Database. It includes information about IPsec security associations, such as SPI, destination addresses, authentication and encryption algorithms and keys.

- PAD. Peer Authorization Database. It provides the link between the SPD and a security association management protocol such as IKE or our SDN-based solution.

4. Objectives

- Flow-based data protection: controller-based flow protection services based on IPsec to allow the protection of specific data flows based on defined security policies.

- Establishment and management of IPsec security associations: this service allows the centralized management of IPsec SAs to protect specific data flows.

5. Case 1: IKE/IPsec in the NSF

In this case, the security controller is in charge of controlling and applying SPD and PAD entries in the NSF. It also has to apply IKE configuration parameters and derive and deliver IKE credentials (e.g. a pre-shared key) to the NSF for the IKE negotiation. In short, we would call this IKE credential.

With these entries and credentials, the IKE implementation can operate to establish the IPsec SAs. The application (administrator) will send the IPsec requirements and end points information, and the security controller will translate those requirements into SPD entries that will be installed in the NSF. With that information provisioned in the NSF, when the data flow needs to be protected, the NSF can just run IKE to establish the required IPsec SA. Figure 1 shows the different layers and corresponding functionality.

Advantages: It is simple because current gateways typically have an IKE/IPsec implementation.

Disadvantages: IKE implementations need to renegotiate IPsec SAs upon SPD entries changes without restarting IKE daemon.

```
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |   IPsec Management/Orchestration Application| Client or
                |                I2NSF Client                 | App Gateway
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                        |      Client Facing Interface
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    Vendor      |             Application Support             |
    Facing <--->+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Security
    Interface   | IKE Credential and SPD Policies Distribution| Controller
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                        |          NSF Facing Interface
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |                 I2NSF Agent                 |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Network
                |   IKE    |      IPsec(SPD,SAD,PAD)          | Security
                +-------------------------------------------- + Function (NSF)
                |         Data Protection and Forwarding      |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Case 1: IKE/IPsec in the NSF

5.1. Requirements

SDN-based IPsec flow protection services provide dynamic and flexible network resource management to protect data flows among network resources and end users. In order to support this capability in case 1, the following requirements are to be met:

o The NSF MUST implement IKE and IPsec databases: SPD, SAD and PAD. It MUST provide an (southbound) interface to provision SPD and PAD entries, IKE Credentials and to monitor the IPsec databases and IKE implementation. Note that SAD entries are created in runtime by IKE.

o A southbound protocol MUST support sending these SPD and PAD entries, and IKE credentials to the NSF.

o It requires an (northbound) application interface in the security controller allowing the management of IPsec SAs.

- In scenarios where multiple controllers are implicated, SDN-based flow protection service may require a mechanism to discover which security controller is managing a specific NSF.

6. Case 2: IPsec (no IKE) in the NSF

This section describes the referenced architecture to support SDN-based IPsec flow protection where the security controller performs automated key management tasks.

```
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            |   IPsec Management/Orchestration Application| Client or
            |               I2NSF Client                  | App Gateway
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                    |   Client Facing Interface
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Vendor      |             Application Support             |
Facing <--->+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Security
Interface   |         SPD, SAD and PAD Entries Distr.     | Controller
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            |       Key Derivation and Distribution       |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                    |   NSF Facing Interface
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            |                  I2NSF Agent                | Network
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Security
            |               IPSec (SPD,SAD,PAD)           | Function (NSF)
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            |        Data Protection and Forwarding       |
            +---------------------------------------------+
```

Figure 2: Case 2: IPsec (no IKE) in the NSF

As shown in Figure 2, applications for flow protection run on the top of the security controller. When an administrator enforces flow protection policies through an application interface, the security controller translates those requirements into SPD,PAD and SAD entries that will be installed in the NSF.

Advantages: 1) It allows lighter NSFs (no IKE implementation), which benefits the deployment in constrained NSFs. 2) IKE does not need to be run in gateway-to-gateway scenario with a single controller (see Section 10.1).

Disadvantages: The overload of IPsec SA establishment is shifted to the security controller since IKE is not in the NSF. As a consequence, this may result in a more complex implementation in the SDN controller side. For example, the security controller needs to

supervise the IPsec SA rekeying so that, after some period of time (e.g. IPsec SA soft lifetime), to create a new IPsec SA and remove the old one. Another example is the NAT traversal support. In this case, since the security controller has a complete view of the network (as SDN paradigm assumes) it can determine tha there is a NAT between two NSFs and apply the required policies to both NSFs besides activating the usage of UDP encapsulation of ESP packets.

6.1. Requirements

In order to support case 2, the following requirements are to be met:

- It requires the provision of SPD, PAD and SAD entries into the NSF. A southbound protocol MUST support sending this information to the NSF.

- NSF MUST be capable to protect data flows with IPsec, such as the capability to forward data through an IPsec tunnel.

- It requires an (northbound) application interface in the security controller allowing the management of IPsec policies.

- In scenarios where multiple controllers are implicated, SDN-based flow protection service may require a mechanism to discover which security controller is managing a specific NSF.

7. Abstract interface (NSF facing interface)

The cases presented above require an analysis of the communication channel between the IPSec stack and the security controller that is performing the key management operations.

The IETF RFC 2367 (PF_KEYv2) [RFC2367] provides a generic key management API that can be used not only for IPsec but also for other network security services to manage the IPsec SAD. Besides, as an extension to this API, the document [I-D.pfkey-spd] specifies some PF_KEY extensions to maintain the SPD. This API is accessed using sockets.

An I2NSF Agent implementation in the NSF can interact with both APIs in a kernel and returns and provides the same information using the NSF Facing Interface. In the following, we show a summary of these messages just to show an example of what may provide the NSF Facing Interface. The details and the accurate information is in RFC 2367 and [I-D.pfkey-spd].

PF_KEY	PF_NETLINK
SADB_ADD	XFRM_MSG_NEWSA
SADB_GETSPI	XFRM_MSG_ALLOCSPI
SADB_UPDATE	XFRM_MSG_UPDSA
SADB_DELETE	XFRM_MSG_DELSA
SADB_GET	XFRM_MSG_GETSA
SADB_ACQUIRE	XFRM_MSG_ACQUIRE
SADB_REGISTER	Subscribe to NETLINK_XFRM group via bind() on the PF_NETLINK socket
SADB_EXPIRE	XFRM_MSG_EXPIRE
SADB_FLUSH	XFRM_MSG_FLUSHSA
SADB_DUMP	NLM_F_DUMP message for XFRM_MSG_GETSA
SADB_X_SPDSETIDX	XFRM_MSG_NEWPOLICY
SADB_X_SPDUPDATE	XFRM_MSG_UPDPOLICY
SADB_X_SPDADD	XFRM_MSG_NEWPOLICY
SADB_X_SPDDELETE[2]	XFRM_MSG_DELPOLICY; may optionally specify policy id in the index of the xfrm_policy
SADB_X_SPDGET	XFRM_MSG_GETPOLICY
SADB_X_SPDACQUIRE	XFRM_MSG_ACQUIRE
SADB_X_SPDEXPIRE	XFRM_MSG_POLEXPIRE
SADB_X_SPDFLUSH	XFRM_MSG_FLUSHPOLICY

Table 1: PF_KEY to PF_NETLINK mappings

An alternative key management API based on Netlink socket API [RFC3549] is used to configure IPsec on the Linux Operating System. The mappings between the PF_KEY commands used in this document and their PF_NETLINK equivalents is provided in Table 1

To manage the IPsec SAD we have the following messages in the PF_KEYv2 API:

- The SADB_GETSPI message allows a process to obtain a unique SPI value for given security association type, source address, and destination address. This message followed by an SADB_UPDATE is one way to create a security association (SADB_ADD is the other method).

- The SADB_UPDATE message allows a process to update the information in an existing Security Association.

- The SADB_ADD message is nearly identical to the SADB_UPDATE message, except that it does not require a previous call to SADB_GETSPI.

- The SADB_DELETE message causes the kernel to delete an IPsec SA from the SAD.

- The SADB_GET message allows a process to retrieve a copy of a Security Association from the SAD.

- The SADB_ACQUIRE message is typically triggered by an outbound packet that needs security but for which there is no applicable IPsec SA existing in the SAD.

- The SADB_REGISTER message allows (a socket) to receive SADB_ACQUIRE messages for the type of IPsec SA.

- The SADB_EXPIRE message is issued when soft limit or hard limit (lifetime) of a IPsec SA has expired.

- The SADB_FLUSH message causes the kernel to delete all entries in its IPsec SAD.

- The SADB_DUMP message causes to dump the operating system's entire IPsec SAD.

Although it is not a standard, KAME IPsec has defined a set of extensions to PF_KEY in order to handle the SPD [I-D.pfkey-spd]. The extended API offers the addtional extensions:

- The SADB_X_SPDSETIDX message allows a process to add only selector of the security policy entry to the SPD.

- The SADB_X_SPDUPDATE message replaces the parameters of an existing SPD entry.

- The SADB_X_SPDADD is message allows a process to add a new security policy entry to the SPD.

- The SADB_X_SPDDELETE message causes the kernel to delete an entry from the SPD.

- The SADB_X_SPDDELETE2 message nearly identical to the SADB_X_SPDDELETE message, except that it specifies the policy id.

- The SADB_X_SPDGET message is allows a process to retrieve a copy of a security policy entry from the SPD.

- The SADB_X_SPDACQUIRE message is triggered by an outbound packet that needs security policy but for which there is no applicable information existing in the SPD.

- The SADB_X_SPDEXPIRE message is issued when limit of a security policy (SPD entry) has expired.

- The SADB_X_SPDFLUSH message causes the kernel to delete all entries in the IPsec SPD.

- The SADB_DUMP causes the kernel to dump all entries in the IPsec SPD.

Regarding PAD management, we have not found any related extension. However, from the abstract data model defined in Section 8 for the PAD an interface could be designed.

8. Data model

These cases assume a data model representing the information to be exchanged between controller and network resource through the southbound interface. As described before this data model has to include the following information [RFC4301] (authors of this I-D are working now on developing a YANG model for this draft):

Data model for the SDP entries:

- Name

- PFP flags

- Perfect forward secrecy

- Selector list:

  Remote IP addresses(es)

  Local IP addresses(es)

  Flow direction

  Next Layer Protocol

  Local port

  Remote port

  Type code

- Processing:

  Extended sequence number

Sequence overflow

Fragment checking

IP compression

DF bit

DSCP

IPsec protocool (AH/ESP)

Algorithms

Manual SPI

Local tunnel endpoint

Remote tunnel endpoint

Tunnel options

Data model for the SAD entries:

- SPI

- Local peer

- Remote peer

- SA mode (tunnel or transport)

- Security protocol

- Sequence number options

- Life-time

- Upper protocol

- Direction

- Tunnel source IP address and port

- Tunnel Destination IP address and port

- AH parameters

- ESP parameters

- IP compression

- NAT traversal flag

- Path MTU

- Anti-replay window

Data model for the PAD entries:

- Identifies the peers or groups of peers that are authorized to communicate with this IPsec entity.

- The protocol and method used to authenticate each peer.

- Authentication data for each peer.

- Constraints about the types and values of IDs that can be asserted by a peer with regard to child SA creation.

- Peer gateway location info (e.g., IP address(es) or DNS names).

Data model for the IKE configuration:

- TBD. (NOTE: It may depend on the IKE version)

9. Scaling considerations

For each new NSF that is added, the existing NSFs may need to be updated with peering information to set up tunnel configuration state to the new node. Setting up this state may need additional message exchanges, and the complexity of this message exchange may merit optimization.

10. Use cases examples

This section explains three use cases as examples for the SDN-based IPsec Flow Protection Service.

10.1. Gateway-to-gateway under the same controller

Enterprise A has a headquarter office (HQ) and several branch offices (BO) interconnected through an Internet connection provided by an Internet Service Provider (ISP). This ISP has deployed a SDN-based architecture to provide connectivity to all its clients, including HQ and BOs, so the HQ is provided with a gateway that acts as a router

between Internet and each BO's internal network. The gateway implements our Flow-based NSF.

Now, Enterprise A requires that certain traffic between the HQ and BOs MUST be protected, for example, with confidentiality and integrity. The Enterprise A's administrator has to configure flow protection policies in the ISP's security controller, determining that the traffic among Enterprise A's HQ (HQ A) and each BO MUST be protected.

```
                 +----------------------------------------+
                 |             Security Controller        |
                 |                                        |
         (1)     |   +--------------+ (2)+--------------+ |
 Flow    ----------> |   Translate  |--->| South. Prot. | |
 Protect. Pol.   |   |IPsec Policies|    |              | |
                 |   +--------------+    +--------------+ |
                 |                          |     |       |
                 |                          |     |       |
                 +--------------------------|-----|-------+
                                            |     |
                                            | (3) |
                  |-------------------------+     +---|
 From             V                                   V         To
 HQ A +----------------------+         +----------------------+ BO
  --->|    NSF1              |<=======>|   NSF2               |---->
      |IKE/IPsec(SPD/SAD/PAD)|         |IKE/IPsec(SPD/SAD/PAD)|
      +----------------------+  (4)    +----------------------+
```

Figure 3: Gateway-to-Gateway single controller flow for case 1 .

Figure 3 describes the case 1:

1. The administrator establishes general Flow Protection Policies.

2. The controller generates IKE credentials and translates the policies into SPD and PAD entries.

3. The controller looks for the NSFs involved (NSF1 and NSF2) and inserts the SPD and PAD entries in both NSF1 and NSF2.

4. All packets belonging to the flow that matches the IPsec SPD inserted by the security controller will trigger the IKE negotiation in NSF1 and NSF2 by using the IKE credentials.

In case 2, Flow Protection Policies defined by the administrator are also translated into IPsec SPD entries and inserted into the corresponding NSFs. Besides, SAD entries will be also defined by the

controller and enforced in the NSFs. In this case the execution of IKE is not necessary in the controller, and a Key Derivation function can be used to provide the required cryptographic material for the IPsec SAs. These keys will be also distributed through the southbound interface. Note that it is possible because both NSFs are managed by the same controller.

```
                  +----------------------------------------+
                  |    (1)      Security Controller        |
      Flow Prot. ---------|                                |
      Pol.        |       V                                |
                  |   +-------------+(2)+---------------+  |
                  |   | Key Deriv. &|-->| South. Prot.  |  |
                  |   | Distribution|   |               |  |
                  |   +-------------+   +---------------+  |
                  |                      |     |           |
                  |                      |     |           |
                  +----------------------| --- |-----------+
                                         |     |
                                         | (3) |
                  |----------------------+     +--|
      From        V                               V           To
      HQ A  +------------------+       +------------------+  BO
    ------->|    NSF1          |<=====>|   NSF2           |------->
            |IPsec(SPD/SAD/PAD)|   4)  |IPsec(SPD/SAD/PAD)|
            +------------------+       +------------------+
```

Figure 4: Gateway-to-Gateway single controller flow for case 2.

Figure 4 describes the case 2, when a data packet is sent from HQ A with destination BO :

1. The administrator establishes Flow Protection Policies.

2. The controller translates these policies into IPsec SPD, PAD and SAD entries.

3. The controller looks for the NSFs involved and inserts the these entries in both NSF1 and NSF2 IPsec databases.

4. All packets belonging to the flow are tunneled between NSF1 and NSF2 by using the enforced configuration keys and parameters. No need to run IKE between NSF1 and NSF2.

In general (for case 1 and case 2), this system presents various advantages to the ISP: (i) it allows to create a IPsec SA among two NSFs, with only the application of specific security policies at the application layer. Thus, the ISP can manage all security

associations in a centralized point and with an abstracted view of the network; (ii) All NSFs deployed after the application of the new policies will NOT need to be manually configured, thus allowing its deployment in an automated manner.

10.2. Gateway-to-gateway under different SDN controllers

Two organizations, Enterprise A and Enterprise B, have its headquarters interconnected through an Internet connection provided by different ISPs, called ISP_A and ISP_B. They have deployed a SDN-based architecture to provide Internet connectivity to all its clients, so Enterprise A's headquarters is provisioned with a gateway deployed by ISP_A and Enterprise B's headquarters is provisioned with a gateway deployed by ISP_B.

Now, these organizations require that certain traffic among its headquarters to be protected with confidentiality and integrity, so the ISPs have to configure Flow Protection Policies in their security controllers. Both administrators define Flow Protection Policies in each Security Controller that will end with the translation into SPD and PAD entries and IKE credentials in each NSF so that the specified traffic exchanged among these headquarters will be protected.

```
                +-------------+                   +-------------+
                |   ISP_A's   |                   |   ISP_B's   |
     Flow Prot. |   Security  |<=================>|   Security <--- Flow Prot.
     Pol.     ---> Controller |        (3)        |  Controller |   Pol.
            (1) |             |                   |             |   (2)
                +-------------+                   +-------------+
                     |                                 |
                     | (4)                         (4) |
    From             V                                 V             To
    HQ A  +----------------------+          +----------------------+ BO
   ------>|    NSF1              |<========>|   NSF2               |------->
          |IKE/IPsec(SPD/SAD/PAD)|          |IKE/IPsec(SPD/SAD/PAD)|
          +----------------------+  (5)     +----------------------+
```

Figure 5: Gateway-to-gateway multi controller flow in case 1

On the one hand, case 1, Figure 5 describes the data and control plane communications required when a data packet is sent from Enterprise A's HQ (HQ A) to destination Enterprise B's HQ (HQ B):

1. The administrator A establishes general Flow Protection Policies in ISP_A's Security Controller

2. The administrator B establishes general Flow Protection Policies in ISP_B's Security Controller

3. The ISP_A's security controller realizes that protection is between the NSF1 and NSF2, which is under the control of another security controller (ISP_B's security controller), so it starts negotiations with the other controller to agree on the IPsec SPD policies and IKE credentials for their respective NSFs. NOTE: This may require extensions in the East/West interface.

4. Then, both security controllers enforce the IKE credentials and related parameters and the SPD and PAD entries in their respective NSFs.

5. All packets belonging to the flow that matches the IPsec SPD inserted by the security controller triggers the IKE negotiation between NSF1 and NSF2 by using the enforced configuration keys and parameters.

```
                +--------------+                   +--------------+
                |   ISP_A's    |                   |   ISP_B's    |
         Flow. --->                     IKE?       |           <---- Flow
         Prot.  |   Security   |<=================>|   Security   |  Prot.
         Pol.(1)|  Controller  |        (3)        |  Controller  |  Pol. (2)
                |              |                   |              |
                +--------------+                   +--------------+
                        |                               |
                        | (4)                       (4) |
        From            V                               V                To
        HQ A    +------------------+      (5)       +------------------+ HQ B
         ------>|    NSF1          |<==============>|    NSF2          |---->
                |IPsec(SPD/SAD/PAD)|                |IPsec(SPD/SAD/PAD)|
                +------------------+                +------------------+
```

Figure 6: Gateway-to-gateway multi controller flow in case 2

On the other hand, case 2, Figure 6 describes the data and control plane communications required when a data packet is sent from Enterprise A's HQ (HQ A) to destination Enterprise B's HQ (HQ B):

1. The administrator A establishes general Flow Protection Policies in ISP_A's Security Controller

2. The administrator B establishes general Flow Protection Policies in ISP_B's Security Controller

3. The ISP_A's security controller realizes that traffic between NSF1 and NSF2 MUST be protected. Nevertheless, the controller notices that NSF2 is under the control of another security controller, so it starts negotiations with the other controller to agree on the IPsec SPD, PAD, SAD entries that define the IPsec

SAs. NOTE: It would worth evaluating IKE as the protocol for the the East/West interface in this case.

4. Once the controllers have agreed on key material and the details of the IPsec SA, they both enforce this information into their respective NSFs.

5. Therefore, all packets belonging to the flow are protected between NSF1 and NSF2 by using the enforced configuration keys and parameters.

In general (case 1 and case 2), this system presents various advantages to both ISPs: (i) it allows to create a security association among two network resources across ISPs, from each ISP point of view, only the application of specific Flow Protection Policies at the application layer is needed, so they can manage all security associations in a centralized point and with an abstracted view of the network; (ii) All new resources deployed after the application of the new policies will not need to be manually configured, thus allowing its deployment in an automated manner.

10.3. **Host-to-gateway**

End user is a member of Enterprise A who needs to connect to the HQ's internal network. Enterprise A has deployed a NSF acting as IPsec-based VPN concentrator in its HQ to allow members of the organization to connect to the HQ's internal network in a secure manner.

Traditionally, VPN concentrators are built as appliances, configured manually to authenticate and establish secure associations with incoming end users users, for example, by running IKE to establish an IPsec tunnel. With the SDN-based management of IPsec we can automatize these configurations.

In case 1, as we can see in Figure 7, the administrator configures a Flow Protection Policy in the security controller (1). The controller generates IKE credentials and translates that into SPD and PAD entries and installs them in the corresponding NSF (2). With those policies and IKE credentials, end user and gateway can negotiate IKE.

```
                  +----------------------------------------+
                  |             Security Controller        |
                  |                                        |
          (1)     |   +--------------+    +--------------+ |
      Flow ---------->| Translate    |--->| South. Prot. | |
      Protect. Pol.   |IPsec Policies|    |              | |
                  |   +--------------+    +--------------+ |
                  |                          |             |
                  |                     (2)  |             |
                  +--------------------------|-------------+
                                             |
                                             V
      +----------+                     +-----------------------+
      | End user |                     |    NSF                | To HQ
      | IKE/IPsec|<===================>| IKE/IPsec(SPD/SAD/PAD)|------->
      +----------+          (3)        +-----------------------+
```

Figure 7: Host-to-gateway flow protection in case 1.

In case 2, IKE implementation now resides in the security controller, as we can see in Figure 8. Here, the NSF needs to forward IKE packets to the controller. Therefore, the IKE negotiation is performed by the end user and the security controller (1), being this fact completely transparent for the end user.

Once the IKE negotiation has been successfully completed, the IPsec SA is available in the end user and in the security controller. The IPsec SA information is to be provisioned into the NSF's SAD, SPD and PAD (2). Now the end user and the NSF share key material, thus being able to establish an IPsec tunnel to protect all traffic among them (3).

In general, this feature allows the configuration of network resources such as VPN concentrators as a service, so these could be deployed and disposed as required by policies, such as network load, in an autonomous manner.

```
                           +-----------------------------------+
                           |       Security Controller         |
                           |                                   |
                           |  +-----------+   +-------------+  |
                           |  |  IKE      |-->| South. Prot.|  |
                           |  |           |   |             |  |
                           |  +-----------+   +-------------+  |
                           |        ^             |            |
                           |        |             |            |
                           +--------|-------------|------------+
                                    |             |
                               (1)  |             | (2)
                                    |             V
           +----------+          +--|----------------+
           |          |<------------+                |
           | End user |          |  Gateway          |   To HQ
           | IKE/IPsec|<========>| IPsec(SPD/SAD/PAD)|------->
           +----------+   (3)    +-------------------+
```

Figure 8: Host-to-gateway flow protection in case 2.

One of the main problems of this scenario is that the security controller has to implement IKE and negotiate with the end user. Additionally, it is still unclear the security implications of performing IKE with a different end point than the NSF. Finally, in terms of implementation, the IKE packets should bypass IPsec protection in the NSF and be forwarded to the security controller.

11. Security Considerations

TBD.

12. Acknowledgements

Authors want to thank David Carrel, Yoav Nir, Tero Kivinen, Linda Dunbar, Carlos J. Bernardos, Alejandro Perez-Mendez and Alejandro Abad-Carrascosa for their valuable comments.

13. References

13.1. Normative References

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

13.2. Informative References

[I-D.ietf-i2nsf-framework]
Lopez, E., Lopez, D., Dunbar, L., Strassner, J., Zhuang, X., Parrott, J., Krishnan, R., Durbha, S., Kumar, R., and A. Lohiya, "Framework for Interface to Network Security Functions", draft-ietf-i2nsf-framework-03 (work in progress), August 2016.

[I-D.ietf-i2nsf-problem-and-use-cases]
Hares, S., Dunbar, L., Lopez, D., Zarny, M., and C. Jacquenet, "I2NSF Problem Statement and Use cases", draft-ietf-i2nsf-problem-and-use-cases-02 (work in progress), October 2016.

[I-D.ietf-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-02 (work in progress), October 2016.

[I-D.jeong-i2nsf-sdn-security-services-05]
Jeong, J., Kim, H., Park, J., Ahn, T., and S. Lee, "Software-Defined Networking Based Security Services using Interface to Network Security Functions", draft-jeong-i2nsf-sdn-security-services-05 (work in progress), July 2016.

[I-D.pfkey-spd]
Sakane, S., "PF_KEY Extensions for IPsec Policy Management in KAME Stack", October 2002.

[I-D.tran-ipsecme-yang]
Tran, K., Wang, H., Nagaraj, V., and X. Chen, "Yang Data Model for Internet Protocol Security (IPsec)", draft-tran-ipsecme-yang-01 (work in progress), June 2015.

[ITU-T.X.1252]
"Baseline Identity Management Terms and Definitions", April 2010.

[ITU-T.X.800]
"Security Architecture for Open Systems Interconnection for CCITT Applications", March 1991.

[ITU-T.Y.3300]
"Recommendation ITU-T Y.3300", June 2014.

[netconf-vpn]
Stefan Wallin, "Tutorial: NETCONF and YANG", January 2014.

[ONF-OpenFlow]
ONF, "OpenFlow Switch Specification (Version 1.4.0)", October 2013.

[ONF-SDN-Architecture]
"SDN Architecture", June 2014.

[RFC2367] McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2", RFC 2367, DOI 10.17487/RFC2367, July 1998, <http://www.rfc-editor.org/info/rfc2367>.

[RFC3549] Salim, J., Khosravi, H., Kleen, A., and A. Kuznetsov, "Linux Netlink as an IP Services Protocol", RFC 3549, DOI 10.17487/RFC3549, July 2003, <http://www.rfc-editor.org/info/rfc3549>.

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

[RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <http://www.rfc-editor.org/info/rfc6071>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <http://www.rfc-editor.org/info/rfc7149>.

Authors' Addresses

Rafa Marin-Lopez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 85 01
Email: rafa@um.es

Gabriel Lopez-Millan
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 85 04
Email: gabilm@um.es

Sowmini Varadhan
Oracle
Redwood Shores, CA
USA

Email: sowmini.varadhan@oracle.com