

I2NSF
Internet-Draft
Intended status: Experimental
Expires: January 9, 2017

R. Marin-Lopez
G. Lopez-Millan
University of Murcia
July 8, 2016

Software-Defined Networking (SDN)-based IPsec Flow Protection
draft-abad-i2nsf-sdn-ipsec-flow-protection-00

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	4
3. Terminology	4
4. Objectives	5
5. Case 1: IKE/IPsec in the NSF	5
5.1. Requirements	6
6. Case 2: IPsec (no IKE) in the NSF	7
6.1. Requirements	7
7. Abstract interfaces	8
8. Data model	10
9. Use cases examples	12
9.1. Gateway-to-gateway under the same controller	12
9.2. Gateway-to-gateway under different SDN controllers	15
9.3. Host-to-gateway	17
10. Security Considerations	19
11. Acknowledgements	19
12. References	19
12.1. Normative References	19
12.2. Informative References	20
Authors' Addresses	21

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 grow 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 defined that 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 NETCONF/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 9.1 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 9.2, where the NSFs involved are under the control of different security controllers.

The host-to-gateway scenario described in Section 9.3 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.

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:

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

- o SPD. IPsec Security Policy Database. It includes information about IPsec policies direction (in, out), local and remote addresses, inbound and outbound SAs, etc.
- o SAD. IPsec Security Associations Database. It includes information about IPsec security associations, such as SPI, destination addresses, authentication and encryption algorithms and keys.
- o 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

- o 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.
- o 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.

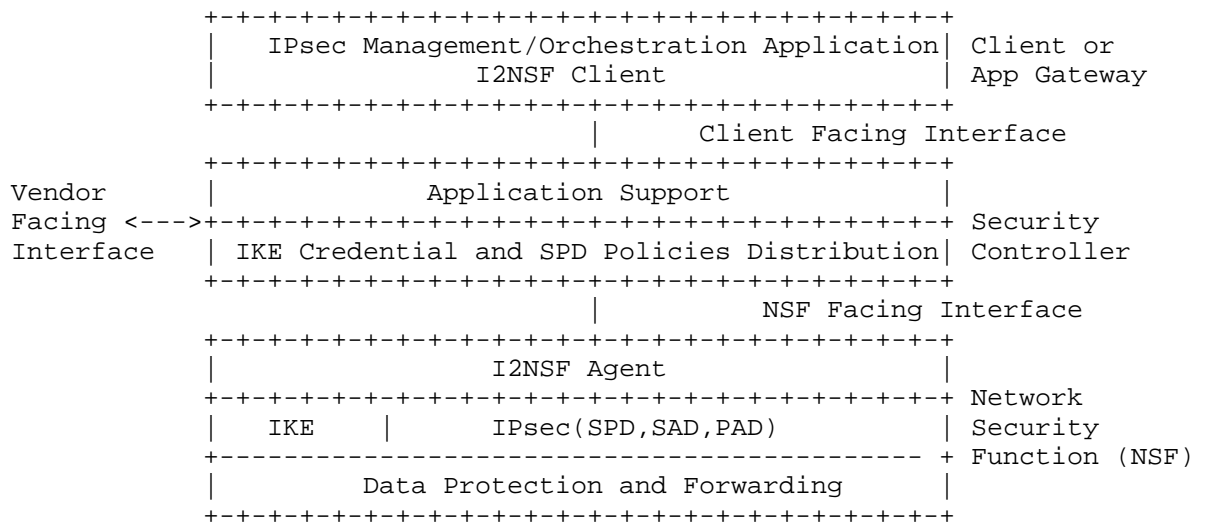


Figure 1: Case 1: IKE/IPsec in the NSF

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.
- o 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.

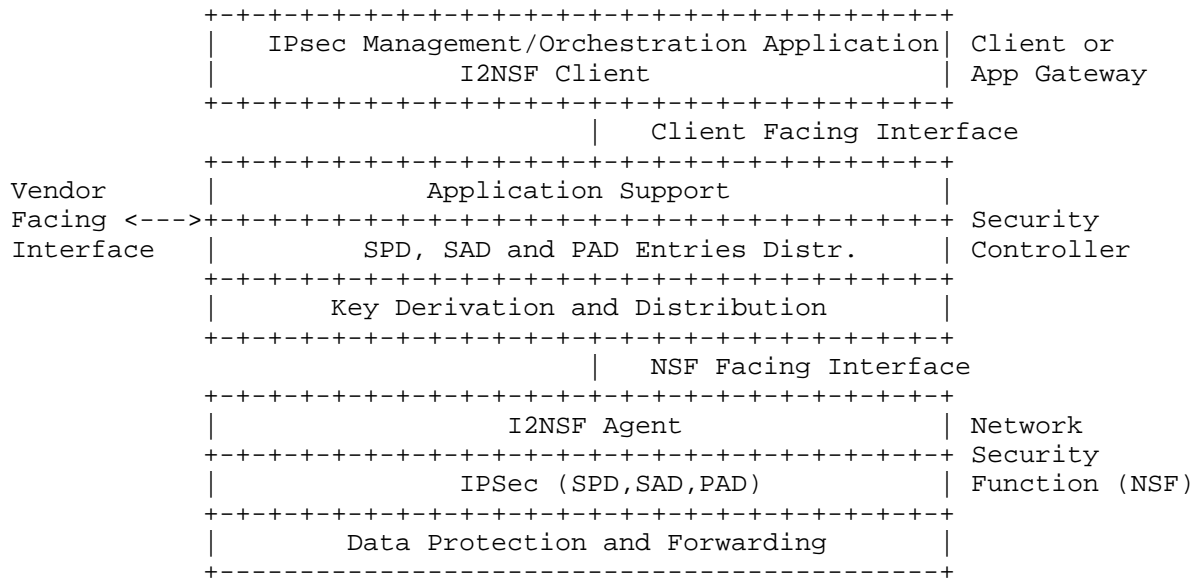


Figure 2: Case 2: IPsec (no IKE) in the NSF

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). 2) IKE does not need to be run in gateway-to-gateway scenario with a single controller (see Section 9.1).

Disadvantages: The overload of IPsec SA establishment is shifted to the security controller since IKE is not required in the NSF.

6.1. Requirements

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

- o It requires the provision of SPD, PAD and SAD entries into the NSF. A southbound protocol MUST support sending this information to the NSF.
- o NSF MUST be capable to protect data flows with IPsec, such as the capability to forward data through an IPsec tunnel.
- o It requires an (northbound) application interface in the security controller allowing the management of IPsec policies.
- o 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 interfaces

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].

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

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

- o The SADB_ADD message is nearly identical to the SADB_UPDATE message, except that it does not require a previous call to SADB_GETSPI.
- o The SADB_DELETE message causes the kernel to delete an IPsec SA from the SAD.
- o The SADB_GET message allows a process to retrieve a copy of a Security Association from the SAD.
- o 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.
- o The SADB_REGISTER message allows (a socket) to receive SADB_ACQUIRE messages for the type of IPsec SA.
- o The SADB_EXPIRE message is issued when soft limit or hard limit (lifetime) of a IPsec SA has expired.
- o The SADB_FLUSH message causes the kernel to delete all entries in its IPsec SAD.
- o 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 additional extensions:

- o The SADB_X_SPDSETIDX message allows a process to add only selector of the security policy entry to the SPD.
- o The SADB_X_SPDUPDATE message replaces the parameters of an existing SPD entry.
- o The SADB_X_SPDADD is message allows a process to add a new security policy entry to the SPD.
- o The SADB_X_SPDDELETE message causes the kernel to delete an entry from the SPD.
- o The SADB_X_SPDDELETE2 message nearly identical to the SADB_X_SPDDELETE message, except that it specifies the policy id.
- o The SADB_X_SPDGET message is allows a process to retrieve a copy of a security policy entry from the SPD.

- o 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.
- o The SADB_X_SPDEXPIRE message is issued when limit of a security policy (SPD entry) has expired.
- o The SADB_X_SPDFLUSH message causes the kernel to delete all entries in the IPsec SPD.
- o 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] (sketch that needs to be developed):

Data model for the SDP entries:

- o Name
- o PFP flags
- o Perfect forward secrecy
- o Selector list:
 - Remote IP addresses(es)
 - Local IP addresses(es)
 - Flow direction
 - Next Layer Protocol
 - Local port
 - Remote port
 - Type code

- o Processing:
 - Extended sequence number
 - Sequence overflow
 - Fragment checking
 - IP compression
 - DF bit
 - DSCP
 - IPsec protocol (AH/ESP)
 - Algorithms
 - Manual SPI
 - Local tunnel endpoint
 - Remote tunnel endpoint
 - Tunnel options

Data model for the SAD entries:

- o SPI
- o Local peer
- o Remote peer
- o SA mode (tunnel or transport)
- o Security protocol
- o Sequence number options
- o Life-time
- o Upper protocol
- o Direction
- o Tunnel source IP address and port

- o Tunnel Destination IP address and port
- o AH parameters
- o ESP parameters
- o IP compression
- o NAT traversal flag
- o Path MTU
- o Anti-replay window

Data model for the PAD entries:

- o Identifies the peers or groups of peers that are authorized to communicate with this IPsec entity.
- o The protocol and method used to authenticate each peer.
- o Authentication data for each peer.
- o Constraints about the types and values of IDs that can be asserted by a peer with regard to child SA creation.
- o Peer gateway location info (e.g., IP address(es) or DNS names).

Data model for the IKE configuration:

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

9. Use cases examples

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

9.1. 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.

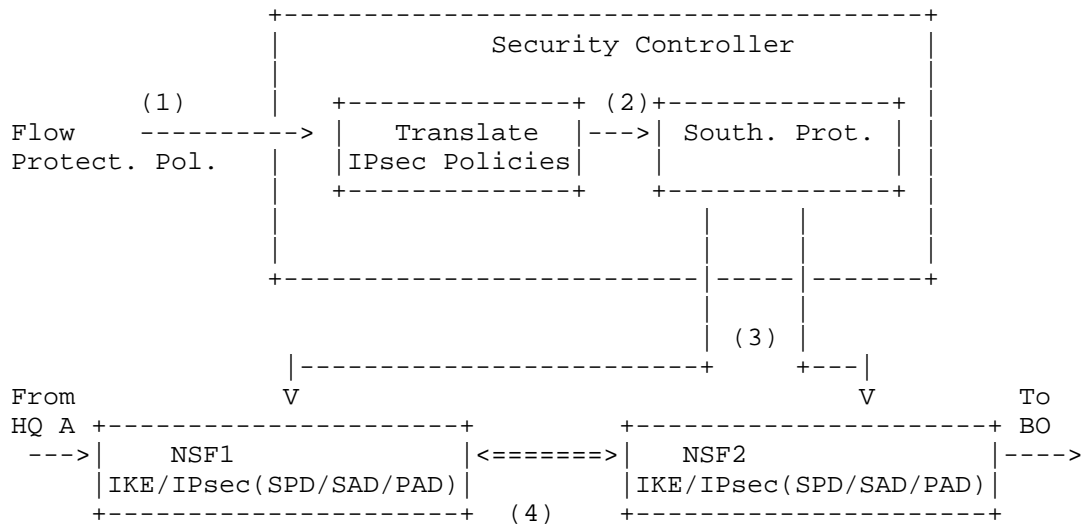


Figure 3: Gateway-to-Gateway single controller flow for case 1 .

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 NSF's are managed by the same controller.

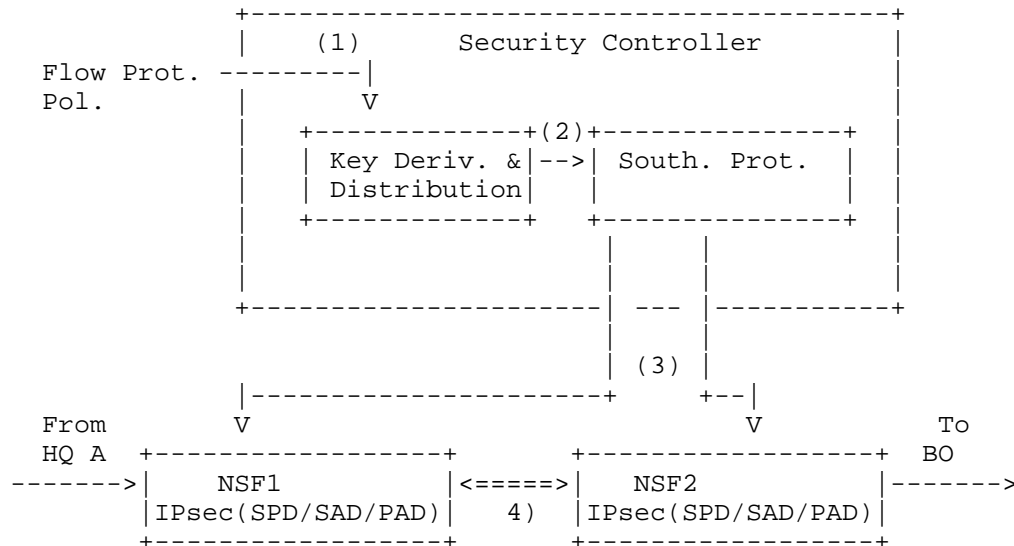


Figure 4: Gateway-to-Gateway single controller flow for case 2.

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 NSF's 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 NSF's, 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 NSF's deployed after the application of the new

policies will NOT need to be manually configured, thus allowing its deployment in an automated manner.

9.2. 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.

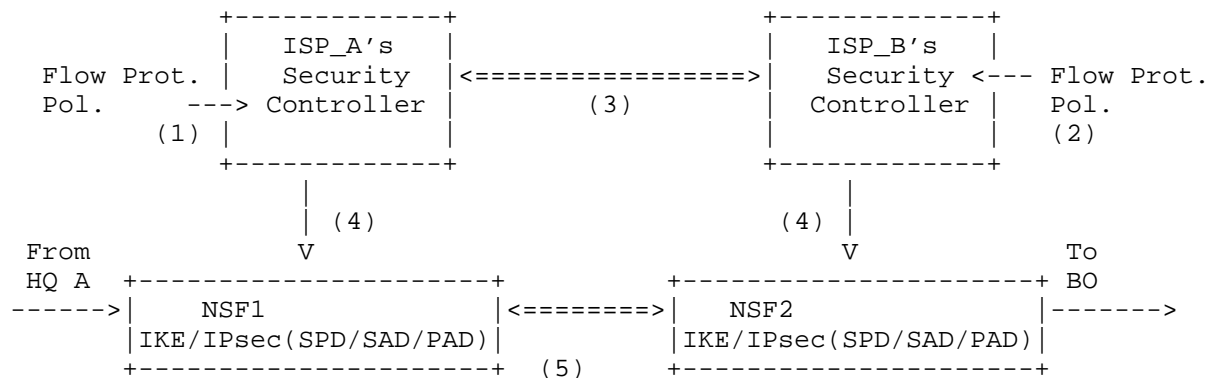


Figure 5: Gateway-to-gateway multi controller flow in case 1

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.

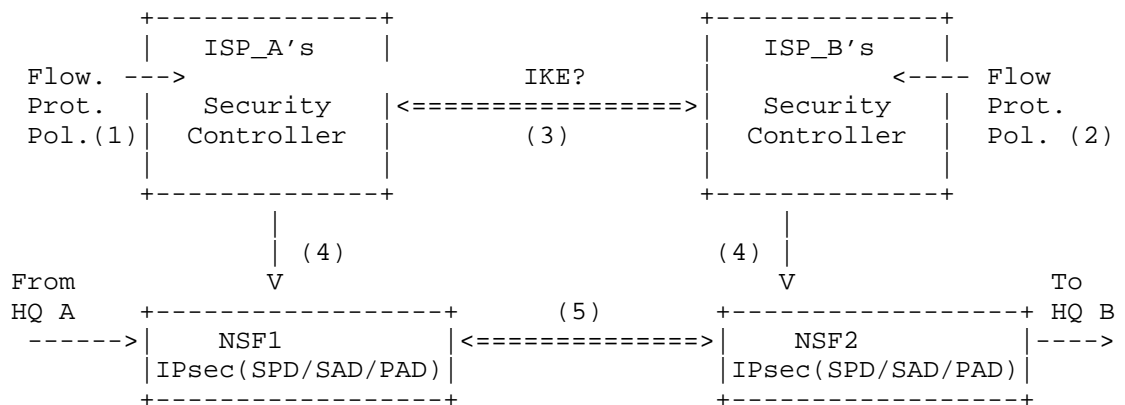


Figure 6: Gateway-to-gateway multi controller flow in case 2

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 be 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.

9.3. 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, 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.

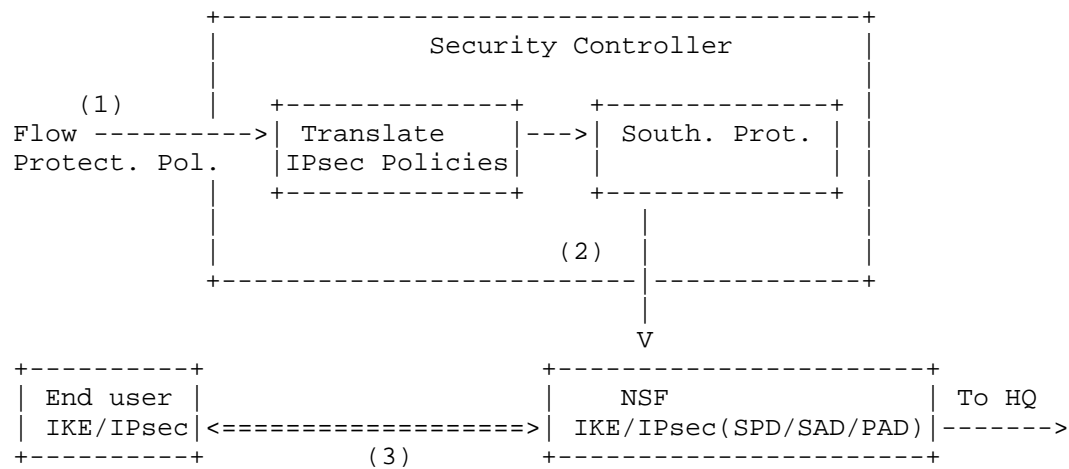


Figure 7: Host-to-gateway flow protection in case 1.

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.

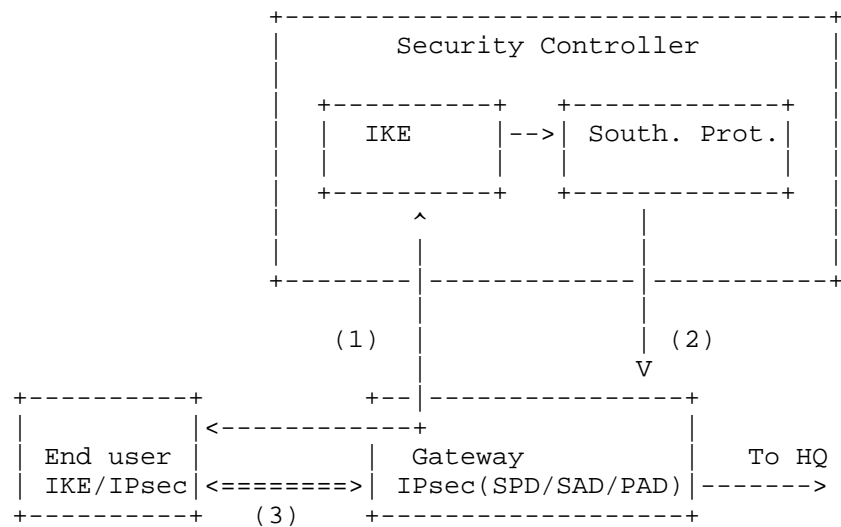


Figure 8: Host-to-gateway flow protection in case 2.

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.

10. Security Considerations

TBD.

11. Acknowledgements

Authors want to thank Sowmini Varadhan, Linda Dunbar, Carlos J. Bernardos, Alejandro Perez-Mendez and Alejandro Abad-Carrascosa for their valuable comments.

12. References

12.1. 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>>.

12.2. Informative References

- [I-D.ietf-i2nsf-framework]
elopez@fortinet.com, e., Lopez, D., Dunbar, L., Strassner, J., Zhuang, X., Parrott, J., Krishnan, R., and S. Durbha, "Framework for Interface to Network Security Functions", draft-ietf-i2nsf-framework-02 (work in progress), July 2016.
- [I-D.ietf-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., and L. Xia, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-00 (work in progress), May 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>>.
- [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