Software-Defined Networking (SDN)-based IPsec Flow Protection
(draft-ietf-i2nsf-sdn-ipsec-flow-protection-02)

Presenter: Rafael Marín López
Gabriel López Millán
(University of Murcia)
SDN-based IPsec

- **Architecture** for the SDN-based IPsec management to centralize the establishment and management of IPsec security associations

- We describe two cases
  - Case 1: When IKEv2 is in the NSF
  - Case 2: When the NSF does not implement IKEv2

- **Goal:** To define the **NSF facing interfaces** required to manage and monitor the IPsec SAs in the NSF from a SC.
  - Case 1) SC provides the NSF with information to IKE, SPD and PAD and can collect state data about IKEv2 and SAD (IPsec SAs)
  - Case 2) SC provides the NSF with valid entries in the SPD and SAD and can collect state about SAD (IPsec SAs)

- **Definition of YANG models for IKEv2, SPD, SAD and PAD**
Case 1 and Case 2

Case 1: IKEv2 in the NSF

Case 2: No IKEv2 in the NSF
YANG model

• The model is based on RFC 4301, RFC 7296 (IKEv2). We have also included some information observed in XFRM API.

• Case 1:
  – IKEv2: it allows to send phase 1 info but phase 2 info is collected from the other containers (PAD, SPD)
  – PAD: it has not changed from previous versions.
  – SPD: to include IPsec policies and read some state date
  – SAD: to collect state data

• Case 2:
  – SPD: to include IPsec policies and collect state data
  – SAD: to configure and collect state date about IPsec SAs
Update (Changes in ietf-...-01)

• New update in section 5.3. Case 1 vs Case 2 discussion
  – Describing rekeying process in more detail
  – NSF state loss
  – NAT traversal behavior

• Added state date to YANG model
  – IKEv2: NAT activated, running since, childs SAs’ SPIs
  – SAD: e.g. current IPsec SA lifetime
  – SPD: e.g. current policy lifetime
NAT Traversal

• Case 1: IKEv2 has a mechanism to detect NAT Traversal
• Case 2: It relays on the assumption that Security controller knows the network it controls, and can know (or discover) if the network devices have NAT configured.
Rekey

• Case 1:
  – IKEv2 in the NSF can control rekey based on the lifetime associated to each IPsec SA.

• Case 2:
  1. The SC chooses two random values as SPI for the new inbound SAs: for example, SPIa2 for A and SPIb2 for B. These numbers MUST not be in conflict with any IPsec SA in A or B. Then, the SC creates an inbound SA with SPIa2 in A and another inbound SA in B with SPIb2 in the NSF A and B respectively. It can send this information simultaneously to A and B.
  2. Once the Security Controller receives confirmation from A and B, inbound SA are correctly installed. Then it proceeds to send in parallel to A and B the outbound SAs: it sends the outbound SA to A with SPIb2 and the outbound SA to B with SPIa2. At this point the new IPsec SAs are ready.
  3. The Security Controller deletes the old IPsec SAs from A (inbound SPIa1 and outbound SPIb1) and B (outbound SPIa1 and inbound SPIb1) in parallel.
Implementation

• We have a NSF implementation:
  – Case 1: IKEv2 (strongswan), NETCONF/YANG (netopeer)
  – Case 2: NETCONF/YANG (netopeer)
  – We have been able to provide a basic configuration for the IPsec SAs and IKEv2 using a NETCONF client

• Security controller side:
  – We have explored ODL and ONOS. We have been be able to configure NSFs with both controllers. But it still needs a lot work.

• Goal: a complete proof-of-concept.
To be done

• Review of the YANG model.
  – We already got a Paul Wouter’s review and apply some comments. But we require more.
  – Minor corrections:
    • To include some variable to INITIAL_CONTACT for IKEv2 model
    • Add SAD lifetime that should be applied to IPsec SAs in SPD

• At implementation level:
  – Continue the work in the controller side. We need to complete an autonomous scenario. We would appreciate collaboration in this side.
  – Small deployments
Software-Defined Networking (SDN)-based IPsec Flow Protection
(draft-ietf-i2nsf-sdn-ipsec-flow-protection-02)

Presenter: Rafael Marín-López
Gabriel López-Millán
(University of Murcia)