Software-Defined Networking (SDN)-based IPsec Flow Protection

(draft-ietf-i2nsf-sdn-ipsec-flow-protection-03)

Presenter: Gabriel López-Millán

Rafael Marín-López

(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 about SAD (IPsec SAs)
- Definition of YANG models for IKEv2, SPD, SAD and PAD

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-…-02)

- New update in section 9. Security Considerations
  - Emphasize the necessity of a security association between the SC and the NSFs, …
  - … and the SC SHOULD never store neither authentication (case 1) nor integrity/encryption (case 2) key material
  - Improve description of security consideration for case 2
- YANG model
  - IKEv2 model:
    - bool variable INITIAL_CONTACT for IKEv2 model
    - SAD lifetime that should be applied to IPsec SAs in SPD
      - ipsec-sad-lifetime-hard
      - Ipsec-sad-lifetime-soft

Implementation

- We have a NSF implementation:
  - Source code: https://gitlab.atica.um.es/gabilm.um.es/cfgipsec2
  - Based on NETCONF/YANG (sysrepo/netopeer2)
  - Case 1: IKEv2 (Strongswan), Case 2: Ubuntu (pfkey_v2)
  - We have been able to test:
    - Basic conf. cases 1 and 2 / host-2-host and gw-2-gw scenarios
    - Rekey mechanism described in the draft document
  - SC based on the netopeer-cli> command line tool (XML conf. examples)
  - Testing: https://gitlab.atica.um.es/gabilm.um.es/sysrepo-netopeer2-cfgipsec2
- Security controller side:
  - ODL and ONOS explored. We have been be able to configure NSFs with both controllers. But it still needs a lot work.
  - We are working in a python-based implementation

Next Steps

- We think the document is ready for the WGLC.
- At implementation level:
  - Continue the work in the controller side. We need to complete an autonomous scenario. We would appreciate collaboration in this side.
  - Implement the complete model and test advanced scenarios

Software-Defined Networking (SDN)-based IPsec Flow Protection

(draft-ietf-i2nsf-sdn-ipsec-flow-protection-03)

Presenter: Gabriel López-Millán

Rafael Marín-López

(University of Murcia)

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.