Using MACsec to protect a Network Functions Virtualisation Infrastructure

(POSTER) Using MACsec to protect a Network Functions Virtualisation infrastructure

Antonio Lioy
Politecnico di Torino
Dip. Automatica e Informatica
Torino, Italy
lioy@polito.it

Ignazio Pedone
Politecnico di Torino
Dip. Automatica e Informatica
Torino, Italy
ignazio.pedone@polito.it

Silvia Sisinni
Politecnico di Torino
Dip. Automatica e Informatica
Torino, Italy
silvia.sisinni@polito.it

***Abstract*—IEEE 802.1AE is a standard for Media Access Control security (MACsec), which enables data integrity, authentication, and confidentiality for traffic in a broadcast domain. This protects network communications against attacks at link layer, hence it provides a higher degree of security and flexibility compared to other security protocols, such as IPsec. Softwarised network infrastructures, based on Network Functions Virtualisation (NFV) and Software Defined Networking (SDN), provide higher flexibility than traditional networks. Nonetheless, these networks have a larger attack surface compared to legacy infrastructures based on hardware appliances. In this scenario, communication security is important to ensure that the traffic in a broadcast domain is not intercepted or manipulated. We propose an architecture for centralised management of MACsec-enabled switches in a NFV environment. Moreover, we present a PoC that integrates MACsec in the Open Source MANO NFV framework and we evaluate its performance.**

***Index Terms*—network security, encryption, security management, data security**

I. Introduction

Local Area Networks (LANs) connect nodes in a broadcast domain. Modern infrastructures, such as cloud platforms by *Internet Service Providers* (ISPs), rely on various technologies to partition and isolate the broadcast domains, i.e. *Virtual LANs* (VLANs) and *Virtual Extensible LANs* (VXLANs), so that packets exchanged between instances in the virtual and physical domains are separated. The protocols used to manage these networks run over the networks themselves and they were designed without considering the threats posed by a network attack. An attacker with physical access to the network could capture, modify, and transmit arbitrary data frames, so it is highly desirable to secure the entire network from unauthorized parties attacks. Since it is impractical to secure the physical access to the whole network, we need to protect the network as close as possible to the physical layer.

The IEEE 802.1AE standard [1] defines a L2 security protocol called MACsec (MAC Security) for protecting connectionless data transfer between stations in a LAN, including link layer protocols (e.g. ARP). MACsec offers:

- *connection-less data integrity* (detects if a frame was modified since it was created, transmitted, or stored);
- *data origin authenticity* (confirmation that the source of a frame is who it claims to be);
- *confidentiality* (encryption of the frame payload);
- *replay protection* (against malicious data retransmission);
- *bounded receiving delay* (sets a maximum data delay).

Given its hop-by-hop nature, MACsec allows channel independence of each node connected to a switch.

Network security is paramount in cloud-based systems, such as the *Network Functions Virtualisation* (NFV) domain where users' traffic is managed by *Virtual Network Functions* (VNFs). These infrastructures offer an high degree of flexibility and scalability, thanks to the use of virtualisation and resource orchestration. Nonetheless, they have a larger attack surface due to the interplay between networking, virtualisation, and software vulnerabilities. For this reason, platform protection is mandatory for use in a production environment.

MACsec is compliant with standard Ethernet LAN, VXLAN and other tunnelling technologies, making it practically usable for virtualised environments such as NFV. In this regard, secure communication can be applied to both traffic exchanged between *NFV Infrastructure* (NFVI) compute nodes and VNFs chained together in a *Service Function Chain* (SFC).

In this work, we propose an architecture for automated establishment of MACsec channels between MACsec-aware hosts and switches, tailored for the NFV environment. This allows both the creation of secure channels between end-user stations and switches, and the automated refresh of cryptographic keys required for the secure communication. Then, we discuss a *Proof of Concept* (PoC) for this architecture that exploits standard Linux tools and the ETSI-backed reference NFV framework *Open Source MANO* (OSM) [2].

II. MACsec application to NFV

A. MACsec target use case for a NFV scenario

Modern softwarised infrastructures comprise a mix of SDN and NFV technologies, so that network flows can be configured on-demand in a flexible way. In this scenario, link layer communication is managed by programmable switches whose control plan is centralised in the administrative domain.

We believe that a use case with MACsec-enabled programmable switches offers the most flexible approach to link layer security in a NFV context. Our scenario is depicted in Fig. 1 and represents different VNFs that need to communicate with each other. These instances may be deployed on

978-1-6654-9792-3/22/$31.00 ©2022 IEEE

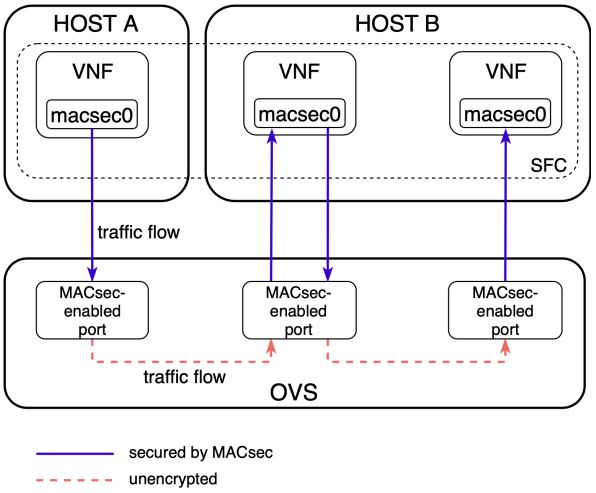

Fig. 1. NFV use case scenario

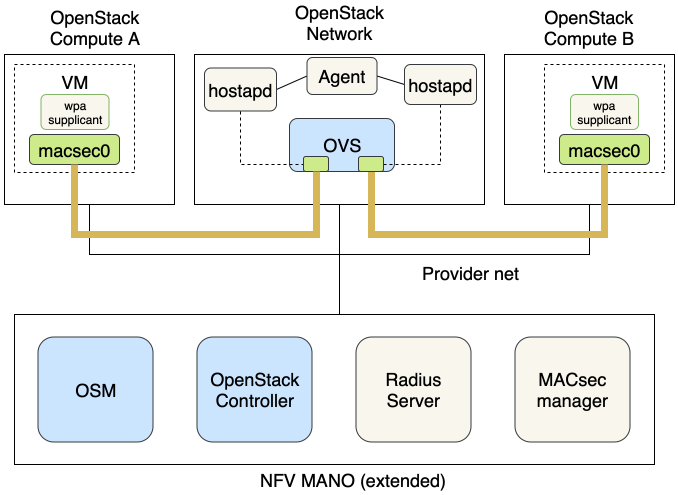

Fig. 2. The MACsec-enabled NFV architecture

different physical nodes and are considered part of a SFC. L2 connectivity is granted by SDN with a set of virtual switches implemented by *Open vSwitch* (OVS) [3]. With respect to security, all flows share the same vulnerabilities: the L2 traffic is in clear hence it is unprotected. Even if higher level protocols (e.g. IPsec, TLS) are used, the data frame is not protected against L2 attacks.

Another layer of complexity is represented by the heterogeneity of *Virtualized Infrastructure Manager* (VIM) platforms. The NFV MANO platform is responsible for the management of both VNF instantiation and provisioning of the connectivity among instances by abstracting the VIM layer. Because of this, we address integration of MACsec management within the MANO administrative domain.

B. MACsec integration in NFV MANO: proposed architecture

Fig. 2 depicts our proposal of enabling MACsec in the reference NFV architectural framework [4]. It is a client-server architecture with a centralised component, the MACsec *Manager*, in charge of managing the instantiation of MACsec channels through an *Agent* running on each NFVI switch.

The MACsec channel is established between the switch (serving as an *authenticator*, implemented via the Hostapd access point daemon) and the host (acting as a *supplicant*, via the *WPA supplicant* tool), after performing the IEEE 802.1X EAP-TLS authentication. We customised these tools to perform key derivation from the EAP *Master Session Key* (EAP-MSK) [5], creation of the key-agreement session, and distribution of the security association key to the supplicant.

Hostapd is also used to manage an OVS switch, providing methods to automatically add and delete ports. The Agent automatically configures Hostapd to connect to a RADIUS [6] server to perform automatic generation of MACsec channels. In our work we integrated FreeRADIUS [7] in the PoC.

The MACsec Manager supports the following operations: list all the registered Agents, and run/list/kill an instance of Hostapd on a registered Agent.

Both the MACsec Manager and the RADIUS server should run in the administrative domain of the NFV infrastructure. We propose to include the MACsec Manager as an extension of the NFV MANO itself. Communication between the MACsec Manager and the Agent is secured by TLS with mutual authentication. Our proposed architecture derives from a practical laboratory use case in which we use OpenStack [8] as Infrastructure-as-a-Service VIM and OSM as NFV *Orchestrator*. As shown in Fig. 2, the architecture comprises four different OpenStack nodes: a controller, two compute nodes, and a networking node. Computation, networking, and storage resources are managed by OpenStack services, such as Nova, Neutron, and Cinder.

The two compute nodes host the instances and run the VNF software. The network node hosts the logic related to the communications (e.g. virtual switches, DHCP services, VXLANs management). Connectivity is managed via OVS, which means that all the virtual networks include OVS bridges.

C. MACsec integration in NFV MANO: channel establishment

The workflow initiator is the NFV Orchestrator whose target is to deploy two (or more) VNF instances connected through a secure channel. This drives the VIM to deploy the instances and to link them to a bridge for connectivity. Afterwards, OSM informs the MACsec manager about which instances and interfaces are involved in the secure channel. The latter contacts the MACsec agent on the network node and starts the Hostapd, which terminates the secure channel started from the instances. Then, the MACsec manager replies with an acknowledgement to the Orchestrator. At this point, the Orchestrator instructs the VNFs to start up the secure channel. Each VNF runs the WPA supplicant, which performs authentication against the Hostapd. The certificates of the MACsec agent and the WPA supplicant modules are verified by the RADIUS server. Moreover, the Agent should be authenticated before the binding with the MACsec manager and the Hostapd creation.

TABLE I
AVERAGE EXECUTION TIMES OF MACSEC WITHOUT AES-NI SUPPORT

security?	*AES-NI?*	`m_start_xmit`	`m_encrypt`	*ratio*
I + C	no AES-NI	20.1 µs	19.5 µs	0.97
I	no AES-NI	12.1 µs	11.6 µs	0.96

TABLE II
AVERAGE EXECUTION TIMES OF MACSEC WITH AES-NI SUPPORT

security?	*AES-NI?*	`m_start_xmit`	`m_encrypt`	*ratio*
I + C	AES-NI	3.2 µs	2.9 µs	0.91
I	AES-NI	2.4 µs	2.2 µs	0.92

Once those steps are completed, the channel between the VNFs and the ports of the bridge is correctly built. Thus, the communication among VNFs is protected against eavesdroppers on the links between both physical and virtual nodes.

III. VALIDATION

A. Test-bed

The validation was performed with the architecture in Fig. 2. OSM R5 is adopted as NFV MANO, configured for deployment of VNFs on OpenStack with two compute instances. Each instance has the following configuration: Intel Core i5-5300U CPU @2.30 GHz, 16 GB RAM DDR 1600 MHz, and 2 NICs 1 Gbps. OSM is hosted on the same node as the OpenStack controller, under a different tenant. The same approach is adopted for the MACsec manager and the RADIUS server. KVM is the virtualisation technology for VNFs, driven by the Nova Compute service. OpenStack VXLANs are managed through standard Linux bridge technology.

B. Encryption overhead

We measured the overhead introduced by MACsec encryption with respect to its transmission latency in two cases: with and without a AES-NI-enabled CPU[1]. We wanted to compute the ratio between the frame transmission time of the MACsec driver (i.e. `macsec_start_xmit`) and the frame encryption time (i.e. `macsec_encrypt`), which is called even when confidentiality is disabled (as integrity is mandatory).

Table I and II show the execution times for both operations and the ratio between them. They have been tested with 1514 B packets in two scenarios: integrity-only and both integrity (I) and confidentiality (C). As expected, there is a performance increase when confidentiality is disabled, but the ratio between the two operations latencies is comparable.

C. End-to-end MACsec throughput

Table III provides the throughput of an end-to-end MACsec channel over Ethernet links at 100 Mbps and 1 Gbps. In this scenario, VNFs are equipped with 4 vCPUs, 6 GB RAM, and 2 NICs based on the *virtIO* para-virtualised drivers [9].

In the case with the 100 Mbps link, the throughput is similar with and without MACsec: 88.42 Mbps versus 90.56 Mbps.

[1]AES-NI is a set of special instructions for Intel architecture to enhance the performance of the AES cryptography.

TABLE III
AVERAGE THROUGHPUT WITH AND WITHOUT MACSEC

Channel bandwidth [Mbps]	*With MACsec* [Mbps]	*Without MACsec* [Mbps]	*loss* [%]
100	88.42	90.56	2.36
1000	488.83	997.75	51

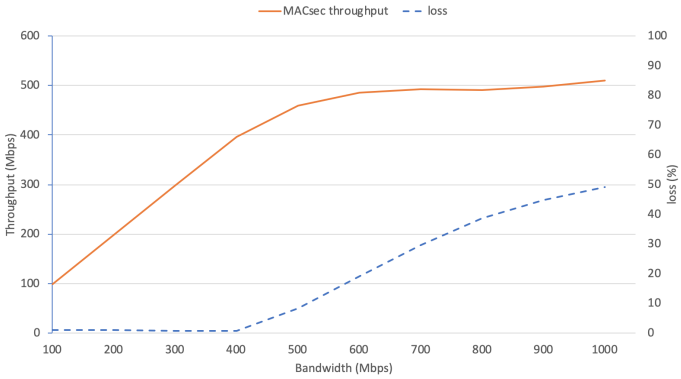

Fig. 3. MACsec throughput variation depending on the channel bandwidth

The performance loss is about 2.4 %, which is a good result considering the added value of data integrity and encryption. With the 1 Gbps channel, the average throughput is 489 Mbps with MACsec, compared to 997 Mbps without MACsec. The loss of 51 % is due to the lack of computing resources, as tests show better results when increasing the number of vCPUs on the VNFs and the virtual switch. Analysis of the window 100 Mbps-1 Gbps shows that the bandwidth increase introduces a loss comparable to the one in the range 0-100 Mbps (Fig. 3). Afterwards, if we keep increasing the maximum bandwidth, we notice a saturation value of the throughput with MACsec enabled. This means that at some point the virtual resources can't keep up with the throughput and need to be enhanced in order to narrow down the gap between the standard case and the one with MACsec.

This evaluation demonstrates that MACsec is a viable approach to secure L2 network communication both on physical and virtual infrastructures. Hence, it can be used for communication between critical instances within a SFC. Nevertheless, significant virtual resources are required to encrypt all the traffic flows in case of high network bandwidth.

REFERENCES

[1] *IEEE Std 802.1AE-2018 - IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Security*, IEEE Std., Dec. 2018. [Online]. Available: https://1.ieee802.org/security/802-1ae/
[2] Open Source MANO website. [Online]. Available: https://osm.etsi.org/
[3] Open vSwitch website. [Online]. Available: https://www.openvswitch.org/
[4] *ETSI GS NFV 002 Network Functions Virtualisation (NFV); Architectural Framework*, ETSI NFV ISG Std., Dec. 2014.
[5] E. P. Aboba B., Simon D., *RFC 5247 - Extensible Authentication Protocol (EAP) Key Management Framework*, IETF Std., 2008.
[6] R. A. Rigney C., Willens S. and S. W., *RFC 2865 - Remote Authentication Dial In User Service (RADIUS)*, IETF Std., 2000.
[7] FreeRADIUS website. [Online]. Available: https://freeradius.org/
[8] OpenStack website. [Online]. Available: https://www.openstack.org/
[9] VirtIO. [Online]. Available: http://www.linux-kvm.org/page/Virtio