January 2020

DYNAMIC, PER-TENANT ENCRYPTION USING SMART NETWORK INTERFACE CARDS

Kyle Mestery

Ian Wells

Grzegorz Duraj

DYNAMIC, PER-TENANT ENCRYPTION USING SMART NETWORK INTERFACE CARDS

AUTHORS:

Kyle Mestery
Ian Wells
Grzegorz Duraj

ABSTRACT

This proposal provides a technique to dynamically encrypt traffic on a per-workload basis on a host from a per-workload virtual switch (vSwitch) to a smart Network Interface Card (NIC). Further, this proposal provides for the ability to utilize the same technique to encrypt traffic between hosts and micro services, thereby facilitating end-to-end encryption of workloads for containerized environments.

DETAILED DESCRIPTION

Smart NICs may allow for the optimization of networking functions. Offloading capabilities of smart NICs may provide advantages in a multi-tenant environment, however, challenges may exist if a network operator desires to make smart NICs multi-tenant in manner such that they can operate with existing tenant traffic.

Consider an example in which a server has a smart NIC and is running applications, in containers or virtual machines, for an N number of tenants. Each tenant may have its own traffic being delivered into the container or virtual machine. The traffic may need to be optimized at a smart NIC level, but encrypting the traffic at the smart NIC level means it is also unencrypted from the smart NIC to a host vSwitch. Thus, the tenant keys must be kept off-box for this encryption.

It may be useful to have a mechanism for encrypting traffic from a smart NIC to a host, especially in high multi-tenancy environments. Solving the problem of key management for this encryption may also be useful.

This proposal provides a technique to dynamically create pre-shared keys (PSKs) for on-host encryption from a per-tenant workload behind a vSwitch to a smart NIC. The smart NIC can create and manage the keys. The smart NIC may also provide an

Application Programming Interface (API) for an orchestrator to share the keys off-host, as well to share encrypted traffic off the smart NIC. The keys may be managed in smart NICs themselves.

Consider, for example, a host setup as shown in Figures 1–4, below, in which a host may be provided with multiple smart NICs and an orchestrator may be provided external to the host.

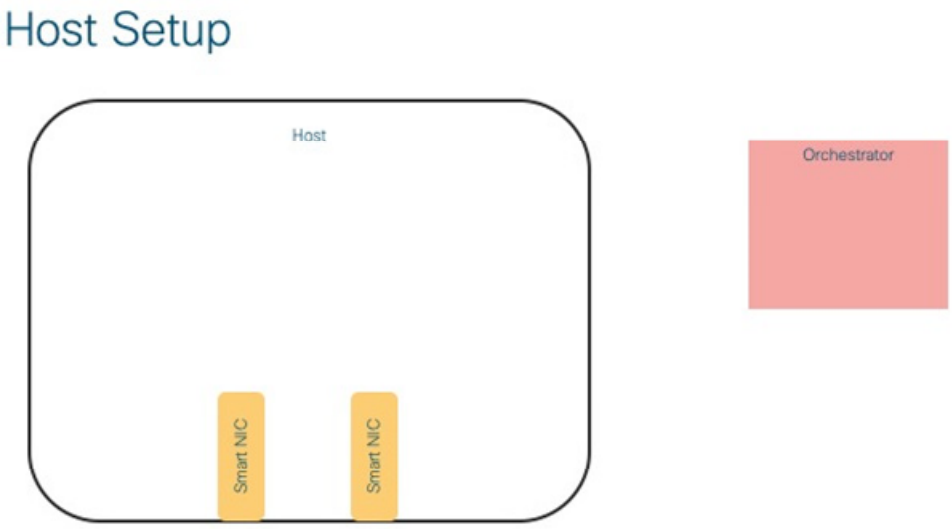

Figure 1

For the host setup shown in Figure 1, the host may be used to run tenant workloads in which the workloads may run in containers inside a namespace. Each workload will have their own vSwitch for traffic communications. The vSwitch may be a kernel based vSwitch or a user space vSwitch.

The orchestration platform will spin-up a tenant workload on the host, as shown in Figure 2, below.

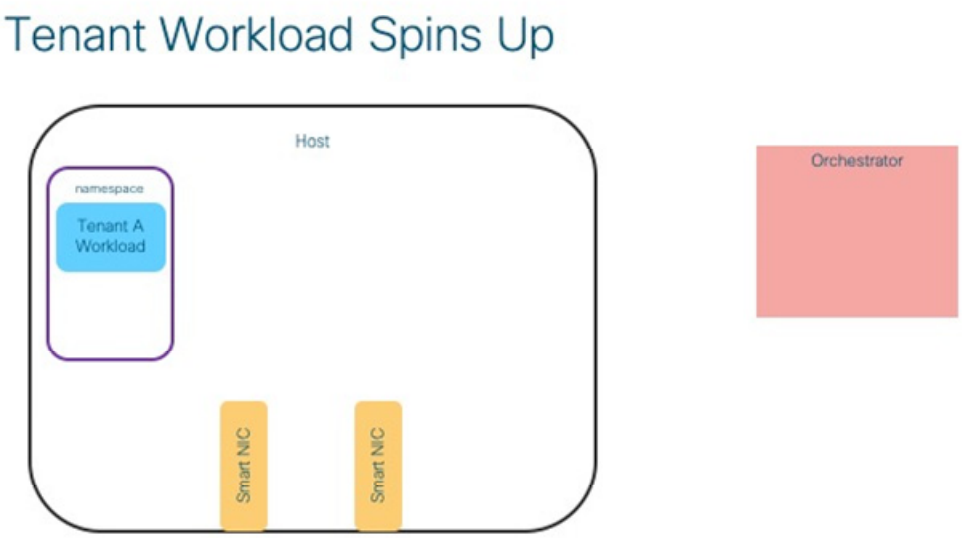

Figure 2

A namespace is created to run the tenant workload in a container. The workload will need an Internet Protocol (IP) address, which will be assigned from the orchestrator into the smart NIC, as shown below in Figure 3.

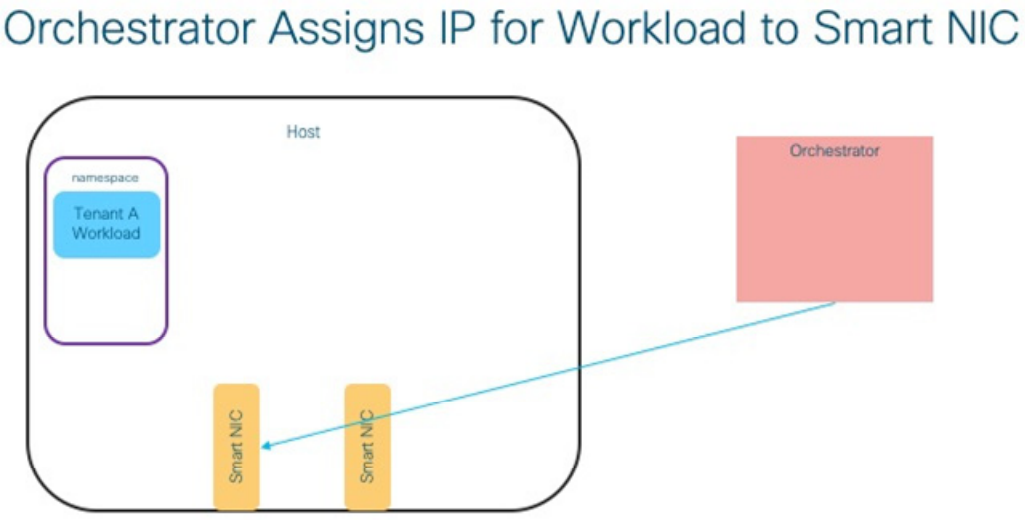

Figure 3

Following IP address assignment, the workload is ready to run and the smart NIC has an IP to hand out (from the orchestrator). The smart NIC may generate a PSK for the encryption algorithm. Note, this could also be a public/private key pair, but such an

implementation would involve the vSwitch also having a public/private key pair. Either a PSK or a public/private key pair may be utilized in accordance with the technique of this proposal. Other encryption techniques may be utilized in accordance with the technique of this proposal such as, for example, Media Access Control security (MACsec), IP security (IPsec), WireGuard®, or the like, as each may provide different advantages. Figure 4, below, illustrates example details associated with a PSK implementation.

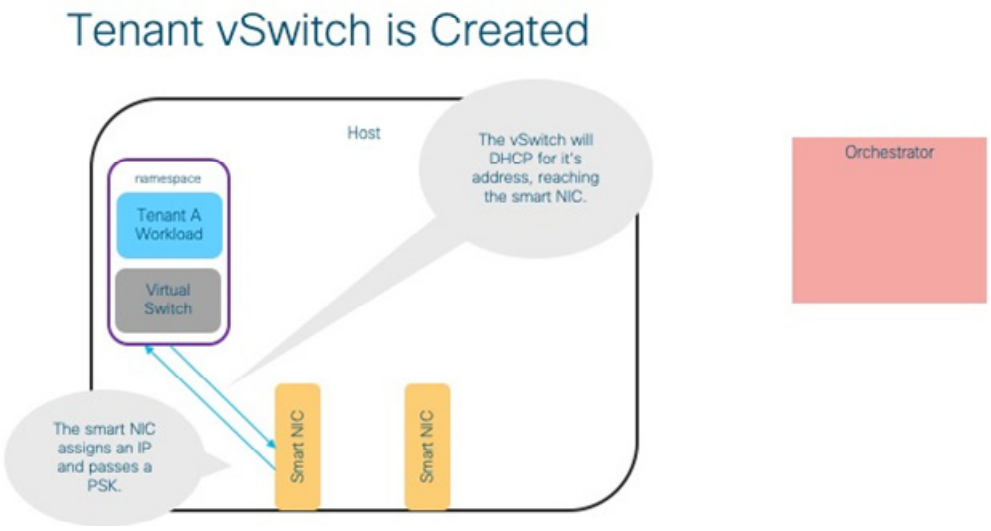

Figure 4

Another instantiation example is shown below in Figures 5–6.

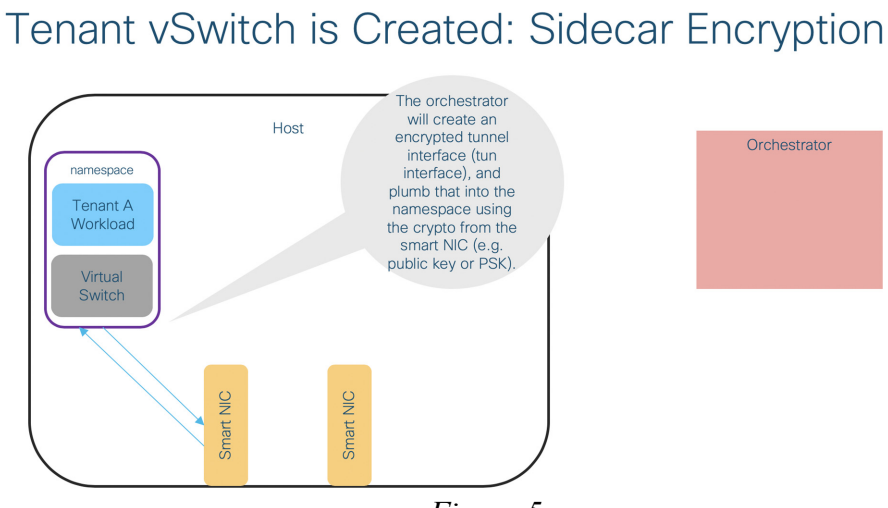

Figure 5

As illustrated in Figure 5, the orchestrator for this implementation can create the workload namespace. The orchestrator may also query the smart NIC over an API to acquire the crypto that is to be utilized. For example, a PSK may be created and shared, a public/private key may be created with the public key that is shared, or any other encryption technique may be utilized.

The smart NIC can map this crypto to the tenant workload and the orchestrator can create the tunnel interface and set up the encryption on that interface. The tunnel interface can be plumbed into the workload namespace and encrypted traffic can flow. In one example, the workload in the namespace can generate normal traffic, and the vSwitch can encrypt the traffic with the PSK (or a public/private key pair) before the traffic egresses the vSwitch and is communicated to the smart NIC, as illustrated in Figure 6, below.

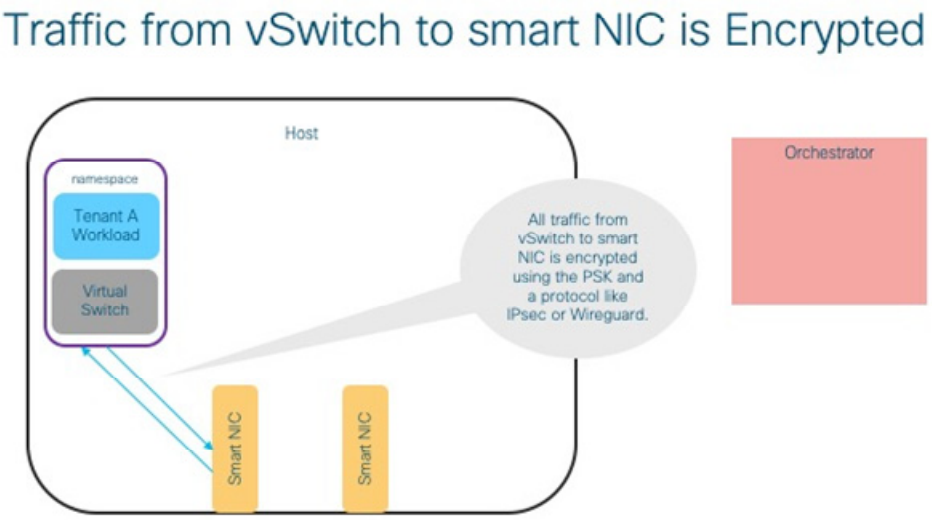

Figure 6

The technique of this proposal ensures that when a host is running more than one workload (as shown in Figure 7, below) each workload is encrypted from the vSwitch to the smart NIC in which each workload may have their own PSK (or public/private key pair) that is generated and stored by the smart NIC.

Multi-Tenant Example

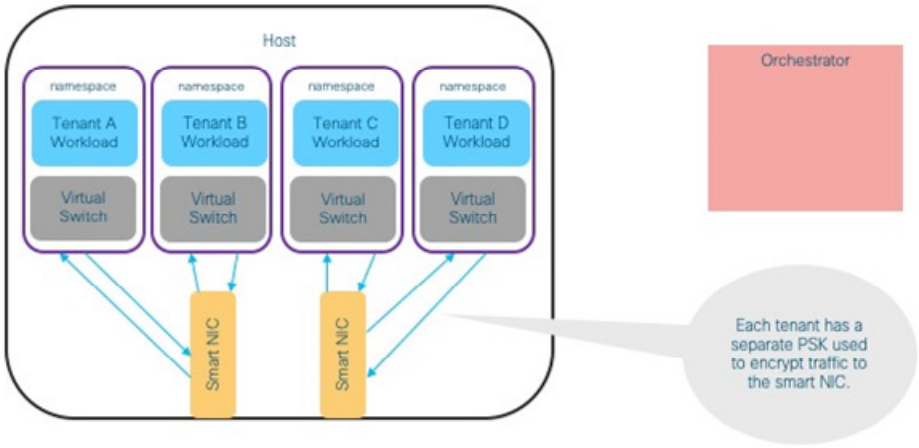

Figure 7

Since the smart NIC has an API, the smart NIC may provide a capability to map PSKs to workloads. As workloads may be spun down, the smart NIC will remove the ephemeral keys. These keys may only exist in the smart NIC memory, rather than being stored on a disk, flash storage, etc.

In some instances, the smart NIC may also be capable of sharing keys via the API, such that the orchestrator may move the workloads between hosts and micro services, which may allow for encrypted traffic off-host, as shown in Figure 8, below.

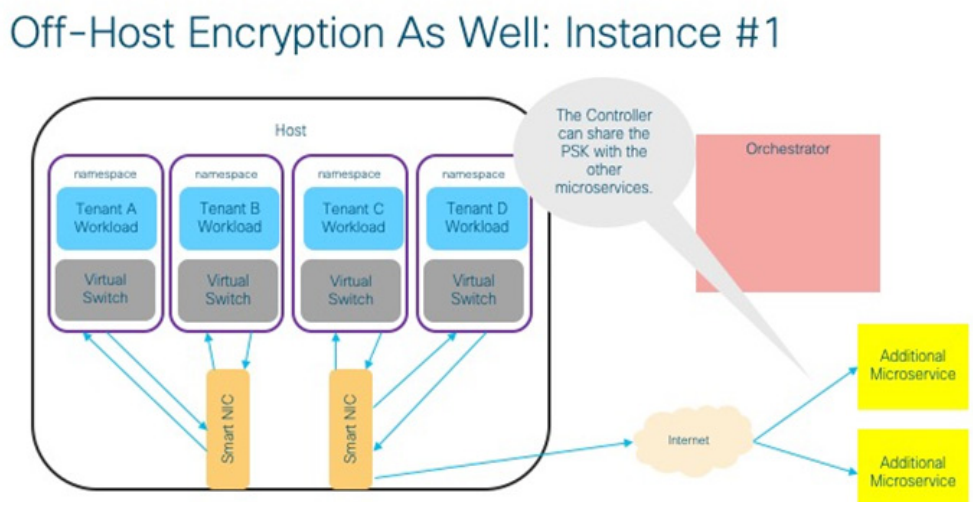

Figure 8

In some implementations, the smart NIC may also generate a separate PSK or key pair for an off-host communication, thereby effectively ensuring that the traffic is encrypted with different keys on-host versus off-host.

Accordingly, this proposal provides for encrypting traffic in both directions using different keys. Traffic from a smart NIC to a host vSwitch is encrypted with keys that can be dynamically generated in the smart NIC itself. These keys may exist only while the host is running. Additionally, traffic from the smart NIC off-host can be encrypted using a different set of keys. This separate keying allows for more granular control of the encryption.

Further, the application running in the workload namespace is not aware that its traffic is encrypted. The encryption is performed automatically underneath and is workload independent, and per-tenant, thereby allowing for isolation from workload to smart NIC and also off-host to another micro service, if applicable.

An important aspect of performing encryption on the smart NIC is that if the smart NIC performs a key exchange (for PKI or otherwise) none of the secrets are passed through the host (which may be contaminated with workload software and/or may not be trusted). Typically, PKI and other split-key systems are used to arrange a shared secret for encrypting and decrypting traffic; it is important that for channels from the host to other locations these secrets do not make it onto the host. Similarly, if non-PKI models are used, such as using a hardware security module (HSM), the shared secret may be delivered directly to the NIC. The secret may be found more widely in the network but still may not find its way onto every host.

In summary, this proposal provides a technique to dynamically encrypt traffic on a per-workload basis on a host from a per-workload vSwitch to a smart NIC. Additionally, this proposal provides for the ability to utilize the same technique to encrypt traffic between hosts and micro services, thereby facilitating end-to-end encryption of workloads for containerized environments.