NIST Special Publication 800-125B

Secure Virtual Network Configuration for Virtual Machine (VM) Protection

Ramaswamy Chandramouli

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-125B

C O M P U T E R S E C U R I T Y

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Special Publication 800-125B

Secure Virtual Network Configuration for Virtual Machine (VM) Protection

Ramaswamy Chandramouli
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-125B

March 2016

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3541 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-125B
Natl. Inst. Stand. Technol. Spec. Publ. 800-125B, 30 pages (March 2016)
CODEN: NSPUE2

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-125B

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

Virtual machines (VMs) are key resources to be protected since they are the compute engines hosting mission-critical applications. Since VMs are end nodes of a virtual network, the configuration of the virtual network is an important element in the security of the VMs and their hosted applications. The virtual network configuration areas discussed in this document are network segmentation, network path redundancy, traffic control using firewalls, and VM traffic monitoring. This document analyzes the configuration options under these areas and presents a corresponding set of recommendations for secure virtual network configuration for VM protection.

Keywords

cloud computing; overlay-based virtual networking; virtual firewall; virtual local area network (VLAN); virtual machine (VM); virtual network segmentation; virtual switch; virtualization

Executive Summary

Data center infrastructures are rapidly becoming virtualized due to increasing deployment of virtualized hosts (also called hypervisor hosts). Virtual machines (VMs) are the key resources to be protected in this virtualized infrastructure since they are the compute engines hosting mission-critical applications of the enterprise. VMs are the end nodes of a virtual network, so the virtual network's configuration is an important element in the overall security strategy for VMs.

The purpose of this NIST Special Publication (SP) is to provide an analysis of various virtual network configuration options for protection of virtual machines (VMs) and present recommendations based on the analysis. The relevant configuration areas discussed in this publication are network segmentation, network path redundancy, traffic control through firewalls, and VM traffic monitoring. Each configuration option in each of these areas has different advantages and disadvantages, which are identified in this publication. Analysis of these has led to the development of one or more security recommendations for each configuration area.

The motivation for this document is the trend in U.S. Federal Government agencies to deploy server virtualization within their internal information technology (IT) infrastructures and to use VMs from cloud service providers for agency applications. The target audience for the document is Chief Information Security Officers (CISOs) and other agency personnel and contractors involved in providing the necessary security protections for agency applications through appropriate virtual network configurations. The intended goal is that the analysis of the configuration options provided in this report, along with the security recommendations, will facilitate making informed decisions with respect to architecting the virtual network configuration. Such a configuration is expected to ensure the appropriate level of network protection for all VMs and the application workloads running in them.

Table of Contents

List of Figures

1. Introduction

A significant trend in the buildup of modern data centers is the increasing deployment of virtualized hosts. A *virtualized host* is a physical host running a server virtualization product (i.e., the hypervisor), making it capable of supporting multiple computing stacks, each with a different platform configuration (e.g., operating system (OS), middleware). The individual computing stack inside a virtualized host (also called a hypervisor host) is encapsulated in an entity called a *virtual machine (VM)*. Since it is a compute engine, a VM has resources assigned to it, such as processors, memory, and storage, and these are called *virtual resources*. A VM computing stack consists of an OS (called the *guest OS*), middleware (optional), and one or more application programs. The application programs loaded into a VM are server programs (e.g., web server, database management system (DBMS)), so the whole process of deploying a virtualized host with one or more VMs running inside it is called *server virtualization*.

A data center with virtualized hosts is said to have a *virtualized infrastructure*. The hypervisor inside each virtualized host can define a network that links its VMs with each other and to the outside (physical) enterprise network. This network is called a *virtual network* since it is entirely software-defined. The core components of this virtual network are one or more virtual network interface cards (vNICs) inside each VM, and virtual switches defined to operate inside the hypervisor kernel. The virtual switches, in turn, are connected to the physical network interface cards (pNICs) of the virtualized host. This provides a communication path for applications and guest OS instances running inside VMs to interact with computing and storage elements on the data center's physical network. The network traffic flowing inside a virtual network can broadly be classified as follows:

- Management traffic (commands for hypervisor administration)
- Infrastructure traffic (e.g., traffic due to VM migration)
- Inter-VM traffic (traffic from applications running in VMs)

The configuration options discussed in this document are applicable for protection of all three types of virtual network traffic listed above, though the focus is on securing inter-VM traffic to protect the VMs and the applications hosted on them.

1.1 Purpose and Scope

As the communication pathway for VMs, each virtual network and its associated configuration parameters play a critical role in ensuring the security of the VMs and the mission-critical applications running inside them. This document discusses four virtual network configuration areas that are of particular interest in terms of security: network segmentation, network path redundancy, traffic control using firewalls, and VM traffic monitoring. Various configuration options in each of these areas have distinct advantages and disadvantages. The purpose of this document is to analyze these advantages and disadvantages from a security viewpoint and provide recommendations to organizations for using the configuration options.

This document only addresses network-level protections for VMs. Two other areas that organizations need to address for ensuring the overall security of the VM and the applications hosted on them are host-level protection and VM data protection. These areas are outside the scope of this document. Most of the host-level protection measures needed for a VM, such as robust authentication and support for secure access protocols (e.g., Secure Shell (SSH)), are no different than the ones needed for physical servers. There are only a few host-level operations specific to VMs that need secure practices (e.g., restarting VMs from snapshots). VM data, generally stored under well-established storage networking technologies (e.g., iSCSI, Fiber Channel), requires protection measures such as encryption, access control and backup schemes and these are outside the scope of this document as well.

1.2 Organization of this Publication

The organization of the rest of this publication is as follows:

- Section 2 discusses network segmentation configuration approaches for VM protection.
- Section 3 examines the options for establishing network path redundancy for VMs.
- Section 4 explores how different types of firewalls can be used to control virtual network traffic.
- Section 5 explains how all incoming and outgoing VM traffic can be captured and monitored.
- Section 6 provides a brief summary for the publication.
- Appendix A lists all the acronyms used in the publication.
- Appendix B offers a bibliography of selected materials related to the topic of this publication.

2. Network Segmentation Configurations for VM Protection

Network segmentation is an often-misunderstood technique. Some security practitioners view network segmentation as being purely for network management purposes. However, other security practitioners consider network segmentation as an integral part or at least a preliminary step of a defense-in-depth network security strategy. Security standards such as the Payment Card Industry Data Security Standard (PCI DSS) specifically cite network segmentation as a requirement for data protection. Similarly, network segmentation is sometimes viewed as being synonymous with use of a virtual local area network (VLAN), but this is inaccurate. This section identifies five distinct approaches to network segmentation, with VLAN deployment being one of them.

The main motivation for network segmentation is to achieve logical separation for applications with different sensitivity levels or belonging to different departments. The network segmentation approaches discussed in this section are organized by their increasing order of scalability. The initial approach is to host all applications of a given sensitivity level in one VM and host all VMs of the same sensitivity level (based on hosted applications) in a given virtualized host (Section 2.1). This is not strictly a network segmentation approach, since it does not involve configuring a network parameter, but it is still included here because it meets the objective of providing VM protection. Sections 2.2 and 2.3 discuss approaches for creating virtual network segments inside a virtualized host using virtual switches and virtual firewalls, respectively. Truly scalable (data center-wide) approaches for creating virtual network segments that span multiple virtualized hosts are discussed in Sections 2.4 and 2.5. These approaches are based on VLAN and overlay-based virtual networking technologies, respectively.

2.1 Separating Virtualized Hosts

When enterprise applications of different sensitivity levels were first being hosted in VMs, the initial network-based protection measure was to locate applications with different sensitivity levels in different virtualized hosts. This isolation between applications was extended into the physical network of the data center by connecting these virtualized hosts to different physical switches and regulating the traffic between these physical switches using firewall rules. Alternatively, virtualized hosts carrying application workloads of different sensitivity levels were mounted in different racks so that they were connected to different top-of-rack (ToR) switches.

2.1.1 Advantages

The most obvious advantage of this approach to segmentation is the simplicity of network configuration and the ease of network monitoring since traffic flowing into and out of VMs hosting workloads of different sensitivity levels are physically isolated.

2.1.2 Disadvantages

The basic economic goal of full hardware utilization is unlikely to be realized when this approach is used because only VMs with the appropriate sensitivity level may be placed onto a particular virtualized host. This tends to negatively impact workload balancing for the data center as a whole. This approach will also hamper flexibility in VM migration because the target host must be of the same sensitivity level as the source host.

2.2 Using Virtual Switches

Another approach to segmentation is to connect VMs with different sensitivity levels to different virtual switches within a single virtualized host. The isolation of traffic between VMs of different sensitivity levels can be achieved by connecting the appropriate virtual and physical switches to each other with their traffic going through one of the virtualized host's pNICs. Finally, the traffic flow between the physical switches has to be regulated through the usual mechanisms, such as a firewall.

2.2.1 Advantages

Segmenting VMs using virtual switches as opposed to hosting them in different virtualized hosts promotes better utilization of virtualized host resources while still maintaining ease of configuration. Further, by design, all hypervisor architectures prevent direct connections between virtual switches within a hypervisor platform, thus providing the necessary isolation.

2.2.2 Disadvantages

There should be as many virtual switches in a virtualized host as there are applications/workloads of different sensitivity levels; this may lead to inefficient use of virtual switches. The flexibility in VM migration may be hampered due to non-availability of ports in the virtual switches of the same sensitivity level (based on the sensitivity level of the migrating VM) in the target host.

2.2.3 Distributed Virtual Switches

The constraints and limitations on VM migration outlined in Sections 2.2.2 and 2.3.2, as well as the network span limitation outlined in Section 2.3.2, do not exist if distributed virtual switches are deployed in the virtualized infrastructure instead of standalone virtual switches. A distributed virtual switch abstracts many individual, host-level virtual switches into a single large virtual switch that spans multiple hosts. Consequently, port groups associated with virtual switches become distributed virtual port groups that span virtualized hosts, thus ensuring consistent virtual network configuration across all hosts, especially those within the same cluster.

2.3 Using Virtual Firewalls

When Internet-facing applications (especially web applications) are run on (non-virtualized) physical hosts, a separate subnet called a Demilitarized Zone (DMZ) is usually created using physical firewalls. Similarly, when VMs hosting web servers running Internet-facing applications are deployed on a virtualized host, they can be isolated and run in a virtual network segment that is separated from a virtual network segment that is connected to the enterprise's internal network. Just as two firewalls – one facing the internet and the other protecting the internal network – are used in a physical network, two firewalls are used inside a virtualized host to create the virtual network equivalent of a DMZ. The major difference in the case of a virtualized host is that the firewalls have to run in a virtual network, so they are virtual software-based firewalls run on dedicated (usually hardened) VMs. A configuration for a DMZ inside a virtualized host is shown in Figure 1.

Figure 1 shows three virtual switches, VS-1, VS-2, and VS-3, inside the virtualized host. The uplink port of VS-1 is connected to the physical NIC pNIC-1, which is connected to a physical switch in the external network. Similarly, the uplink port of VS-3 is connected to the physical NIC pNIC-2, which is connected to a physical switch in the data center's internal network. The firewall appliances running in VM1 and VM4 play the roles of Internet-facing firewall and internal firewall, respectively. VM1 acts as the traffic

control bridge between the virtual switches VS-1 and VS-2, while VM4 acts as the traffic control bridge between the virtual switches VS-2 and VS-3. This configuration creates an isolated virtual network segment based on the virtual switch VS-2 (DMZ of the virtual network), since VS-2 can only communicate with the Internet using the firewall in VM1 and with the internal network using the firewall in VM4. All VMs connected to the virtual switch VS-2 (VM2 and VM3) run in this isolated virtual network segment as well. All traffic involving them and the external network is controlled by the firewall in VM1, and all traffic involving them and the internal network is controlled by the firewall in VM4.

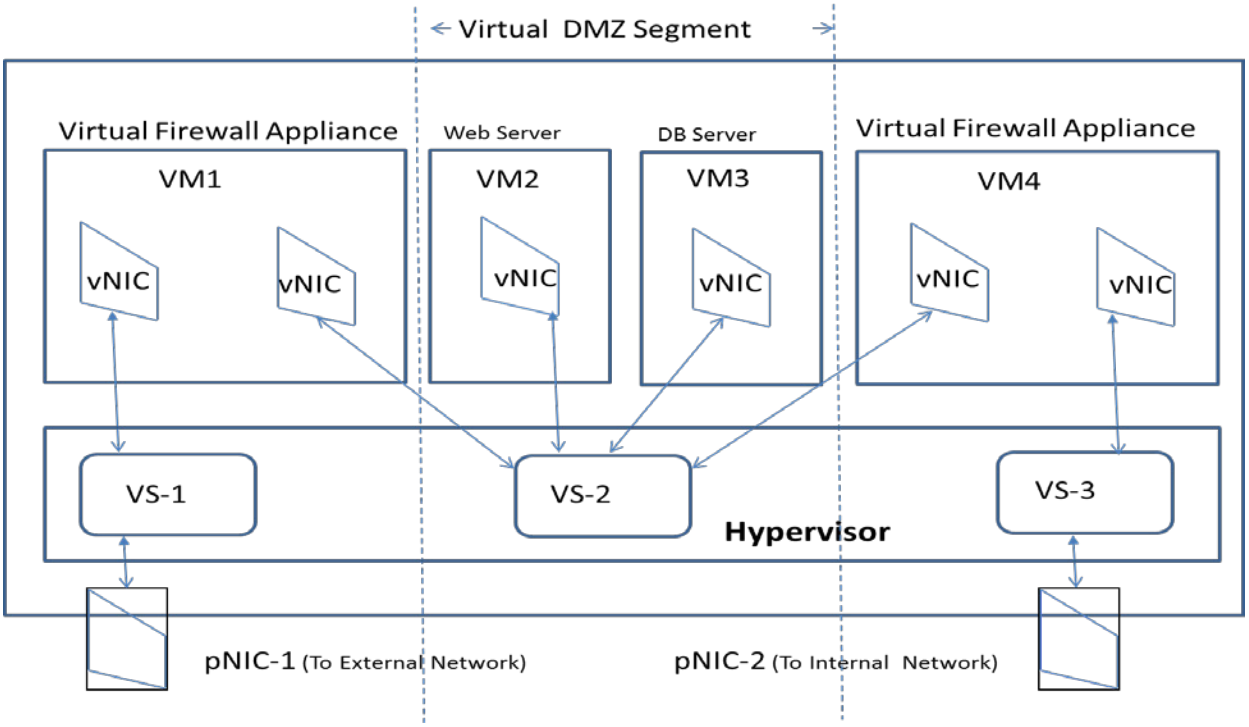

Figure 1: Segmentation Using Virtual Switches and Virtual Firewalls

Looking at the above virtual network configuration from a VM point of view (ignoring whether they run a firewall or a business application), VM1 and VM4 are multi-homed VMs with at least one vNIC connected to a virtual switch that has its uplink port connected to a physical NIC. By contrast, VM2 and VM3 are connected only to an internal virtual switch, VS2, which is not connected to any physical NIC. Such a virtual switch is called an *internal-only switch*. VMs connected only to internal-only switches enjoy a degree of isolation because they run in an isolated virtual network segment.

2.3.1 Advantages

Virtual firewalls come packaged as virtual security appliances on purpose-built VMs, so they are easy to deploy. Also, virtual firewalls can be easily integrated with virtualization management tools/servers, so they can be easily configured (especially their security rules or access control lists (ACLs)) as well.

2.3.2 Disadvantages

The VMs hosting the virtual firewalls compete for the same hypervisor host resources (e.g., CPU, memory) as VMs running business applications, since the former are co-hosted with the latter in the same physical host.

The span of the protected network segment is limited to a single virtualized host. Migration of the VMs in the protected network segment (for load balancing or fault tolerance reasons) to another virtualized host is possible only if the target host has an identical virtual network configuration, so this constrains VM migration flexibility. Creating virtualized hosts with identical virtual network configurations may limit full utilization of the hosts.

2.4 Using VLANs in Virtual Network

VLANs were originally implemented in data centers where nodes were configured to operate in Ethernet-switched modes for ease of control and network management (e.g., broadcast containment). As a network segmentation technique, it provided value as a security measure because of the traffic isolation effect. In a data center with all physical (non-virtualized) hosts, a VLAN is defined by assigning a unique ID called a VLAN tag to one or more ports of a physical switch. All hosts connected to those ports then become members of that VLAN ID, creating a logical grouping of servers (hosts), regardless of their physical locations, in the large flat network of a data center. An example of a VLAN configuration is shown in Figure 2.

The concept of VLANs can be extended and implemented in a data center with virtualized hosts using virtual switches with ports or port groups that support VLAN tagging and processing. In other words, VLAN IDs are assigned to ports of a virtual switch inside a hypervisor kernel, and VMs are assigned to appropriate ports based on their VLAN membership. These VLAN-capable virtual switches can perform VLAN tagging of all packets going out of a VM (with the tag depending upon which port it has received the packet from) and can route an incoming packet with a specific VLAN tag to the appropriate VM by sending it through a port with a VLAN ID assignment equal to the VLAN tag of the packet and with a matching media access control (MAC) address match.

Corresponding to the VLAN configuration of the various virtual switches inside a virtualized host, link aggregation should be configured on links between the pNICs of these virtualized hosts and the physical switches in the data center. This is necessary so that these links can carry traffic corresponding to all VLAN IDs configured inside that virtualized host. Further, the ports of the physical switch, which forms the termination point of these links, should also be configured as trunking ports (capable of receiving and sending traffic belonging to multiple VLANs). A given VLAN ID can be assigned to ports of virtual switches located in multiple virtualized hosts. Thus the combined VLAN configuration, consisting of the configuration inside the virtualized host (assigning VLAN IDs to ports of virtual switches or vNICs of VMs) and the configuration outside the virtualized host (link aggregation and port trunking in physical switches), provides a pathway for VLANs defined in the physical network to be carried into a virtualized host (and vice versa). This provides the ability to isolate traffic emanating from VMs distributed throughout the data center, and thus a means to provide confidentiality and integrity protection to the applications running inside those VMs.

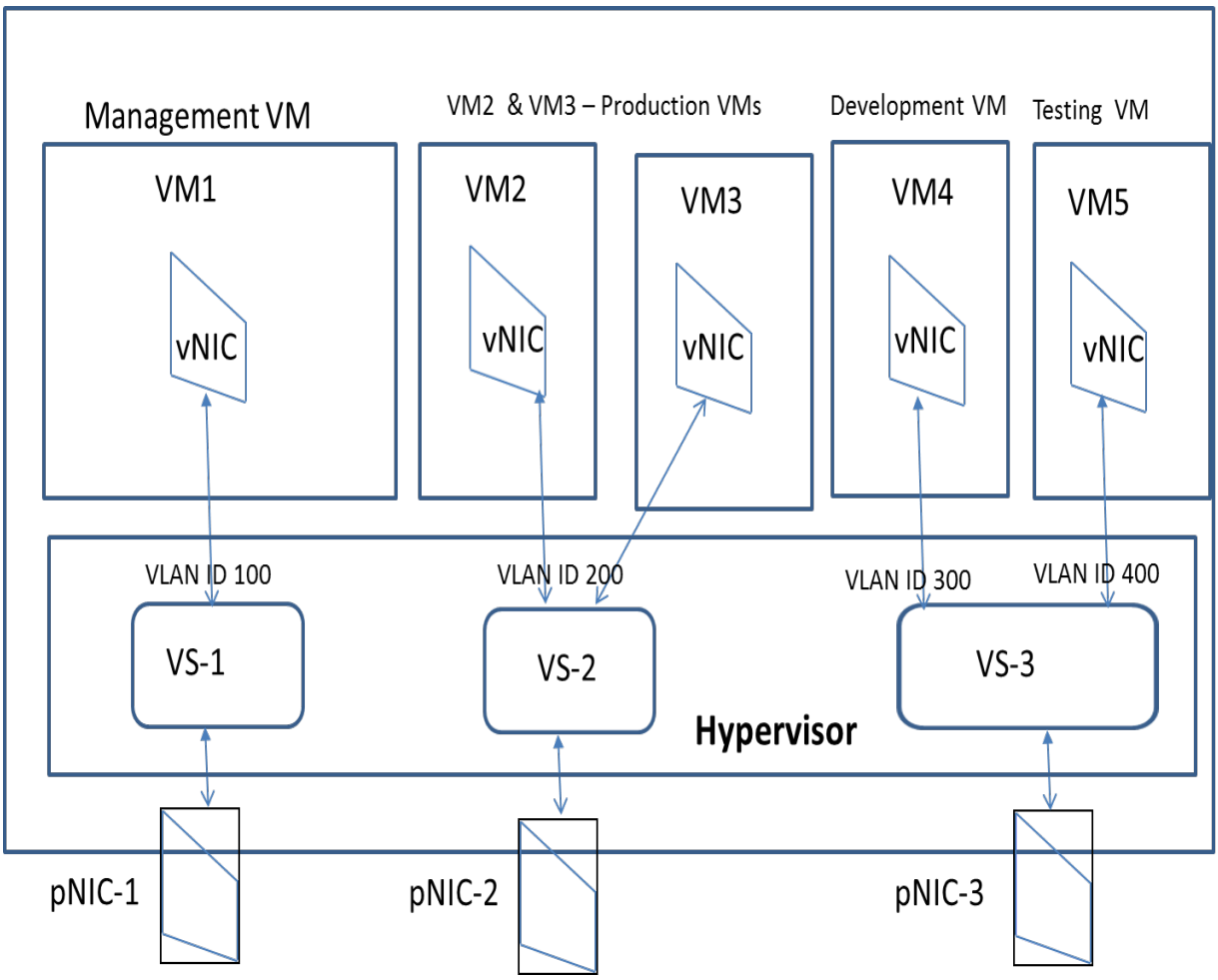

Figure 2: An Example VLAN Configuration

A logical group of VMs is created with the traffic among the members of that group being isolated from traffic belonging to another group. The logical separation of network traffic provided by VLAN configuration can be based on any arbitrary criteria. Thus all of the following can be achieved:

(a) Management VLAN for carrying only management traffic (used for sending management/configuration commands to the hypervisor);
(b) VM Migration VLAN for carrying traffic generated during VM migration (migrating VMs from one virtualized host to another for availability and load balancing reasons);
(c) Logging VLAN for carrying traffic used for fault-tolerant logging;
(d) Storage VLAN for carrying traffic pertaining to Network File System (NFS) or iSCSI storage;
(e) Desktop VLAN for carrying traffic from VMs running Virtual Desktop Infrastructure (VDI) software; and
(f) A set of production VLANs for carrying traffic between the production VMs (the set of VMs hosting the various business applications). These days, enterprise application architectures are made up of three tiers: webserver, application, and database. A separate VLAN can be created for each of these tiers with traffic between them regulated using firewall rules. Further, in a cloud data center, VMs may belong to different consumers or cloud users, and the cloud provider can provide isolation of traffic belonging to different clients using VLAN configuration. In effect, one or more logical or virtual network segments are created for each tenant by assigning/connecting VMs belonging to each of them to different VLAN segments. In

addition to confidentiality and integrity assurances (referred to earlier) provided by logical separation of network traffic, different quality of service (QoS) rules can be applied to different VLANs (depending upon the type of traffic carried), thus providing availability assurance as well.

2.4.1 Advantages

Network segmentation using VLANs is more scalable than approaches using virtual firewalls (Section 2.3). This is due to the following:

- The granularity of VLAN definition is at the port level of a virtual switch. Since each virtual switch can support around 64 ports, the number of network segments that can be defined inside a single virtualized host is much more than what is practically possible using firewall VMs.
- Network segments can extend beyond a single virtualized host (unlike the segment defined using virtual firewalls) since the same VLAN ID can be assigned to ports of virtual switches in different virtualized hosts. The total number of network segments that can be defined in the entire data center is around 4000 (since the VLAN ID is 12 bits long).

2.4.2 Disadvantages

Major disadvantages of achieving network segmentation using VLANs include the following:

- The configuration of the ports in the physical switch (and their links) attached to a virtualized host must exactly match the VLANs defined on the virtual switches inside that virtualized host. This results in tight coupling between the virtual and physical network. The consequence is that the port configuration of the physical switches has to be frequently updated since the VLAN profile of the attached virtualized host may frequently change from VM migrations, VM-hosted application changes, and other reasons. More specifically, the MAC address to VLAN ID mapping in the physical switches may go out of synch, resulting in some packets being flooded through all ports of the physical switch. This results in increased workload on the hypervisors due to processing unnecessary packets.
- The capability to define a network segment spanning virtualized hosts may spur administrators to create a VLAN segment with a large span for providing greater VM mobility (for load balancing and availability reasons). This phenomenon, called *VLAN sprawl*, may result in more broadcast traffic for the data center as a whole, and it also has the potential to introduce a configuration mismatch between the VLAN profile of virtualized hosts and their associated physical switches (discussed earlier).
- The number of VLAN segments that can be defined is limited to just over 4000 (due to the 12-bit namespace for VLAN IDs), so it is not a straightforward process to scale up a VLAN deployment for a large data center.
- There are some production scenarios where VLANs may span across sites. Because of the rigid need to maintain identical VLAN IDs at the multiple sites, VLAN bridging may have to be deployed. VLAN bridging may be easy to configure initially, but maintaining the configuration in response to changes in network profiles may become a costly and/or error-prone process.

2.5 Using Overlay-Based Virtual Networking

In overlay-based virtual networking, isolation is realized by encapsulating an Ethernet frame received from a VM. Various encapsulation schemes (or overlay schemes) can be used, including Virtual Extended

Local Area Network (VXLAN), Generic Routing Encapsulation (GRE), and Stateless Transport Tunneling (STT). Figure 3 illustrates the encapsulation process using VXLAN. First, the Ethernet frame received from a VM, which contains the MAC address of the destination VM, is encapsulated in two stages: first with the 24-bit VXLAN ID (virtual Layer 2 (L2) segment) to which the sending/receiving VM belongs, and second, with the source and destination IP addresses of the VXLAN tunnel endpoints (VTEP), which are kernel modules residing in the hypervisors of the sending and receiving VMs, respectively. VXLAN encapsulation enables creation of a virtual Layer 2 segment that can span not only different virtualized hosts but also IP subnets within the data center.

The two stages of encapsulation used to generate a VXLAN packet are performed by a hypervisor kernel module called the *overlay module*. The overlay module needs the mapping of the MAC address of the remote VM to the corresponding VTEP's IP address. The overlay module can obtain this IP address in two ways: either by flooding using IP learning packets, or by configuring the mapping information using a SDN controller that uses a standard protocol to deliver this mapping table to the overlay modules in each virtualized host. The second approach is more desirable since learning using flooding results in unnecessary network traffic in the entire virtualized infrastructure.

VXLAN-based network segmentation can be configured to provide isolation among resources of multiple tenants of a cloud data center. A particular tenant can be assigned two or more VXLAN segments (or IDs). The tenant can make use of multiple VXLAN segments by assigning VMs hosting each tier (web, application, or database) to the same or different VXLAN segments. If VMs belonging to a client are in different VXLAN segments, selective connectivity can be established among those VXLAN segments belonging to the same tenant through suitable firewall configurations, while communication between VXLAN segments belonging to different tenants can be prohibited.

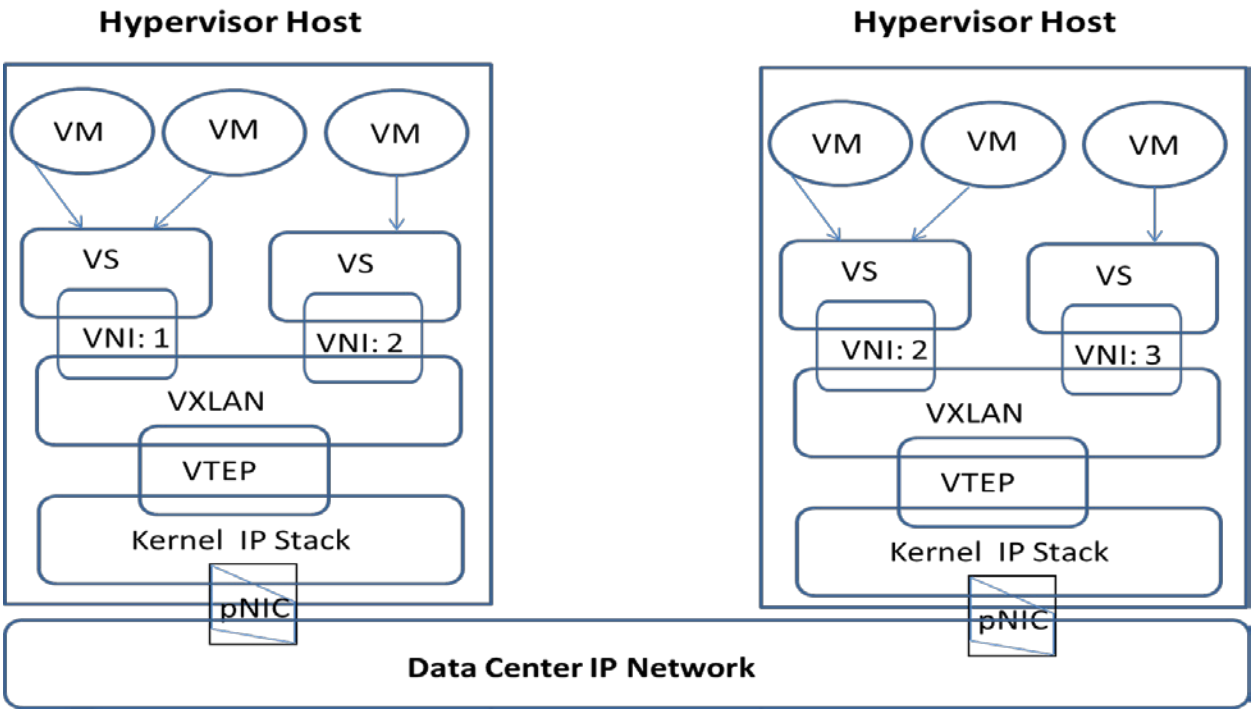

Figure 3: Virtual Network Segmentation using Overlays (VXLAN)

2.5.1 Advantages

Major advantages of overlay-based network segmentation include the following:

- It is much more scalable compared to the VLAN-based approach due to the following:
  (a) A VXLAN network identifier (VNID) is a 24-bit field compared to the 12-bit VLAN ID. Hence the namespace for VXLANs (and the number of network segments that can be created) is about 16 million as opposed to 4096 for VLANs.
  (b) The encapsulating packet for overlay-based network segmentation is an IP/User Datagram Protocol (UDP) packet. So the number of network segments that can be defined is limited only by the number of IP subnets in the data center and not by the number of ports of virtual switches, as is the case for VLAN-based network segmentation.
- In a data center offering Infrastructure as a Service (IaaS) cloud services, isolation between the tenants can be achieved by assigning each of them at least one VXLAN segment (denoted by a unique VXLAN ID). Since VXLAN is a logical L2 layer network running on top of a physical L3 layer (IP) network inside the data center, the latter is independent of the former. In other words, no device of the physical network has its configuration dependent on the configuration of any part of the virtual network. This provides the freedom to locate the computing and/or storage nodes belonging to a particular client in any physical segment of the data center network. In turn, this helps to locate those computing/storage resources based on performance and load balancing considerations. This results in greater VM mobility and availability.
- It eliminates the need to configure the trunking links going into every virtualized host with many VLANs (even though VMs belonging to some of the VLANs may not exist in the host at that time), and thus avoid an increase in traffic due to overprovisioning.
- Any overlay-based deployment in a production environment needs to have a control plane (and hence a controller) that facilitates automation of the provisioning functions, eliminating the chance of errors due to manual provisioning and enabling easier troubleshooting.
- It is easier to configure and manage the physical firewalls since only the VXLAN (or any other overlay scheme) port needs to be allowed for all VM traffic.

2.5.2 Disadvantages

A given network segment (a particular VXLAN ID) can exist in any virtualized host in the data center. This means that routing packets between any two VMs requires large mapping tables for the MAC addresses of the remote VMs and their corresponding VTEP IP addresses. As previously stated, the preferred way to build the mapping tables is to using a centralized or decentralized controller. However, the introduction of this control plane increases the overhead of network management. Another factor to watch for is the size of the mapping table; this can be controlled by carefully limiting the span of overlay segment by proper definition of Layer 2 Transport Zones (as is the case in typical production networks).

2.6 Security Recommendations for Network Segmentation

Based on the analysis of network segmentation approaches for VM protection in Sections 2.1 through 2.5, the following security recommendations are provided. Each recommendation has a unique identifier of format VM-NS-Rx, where VM stands for virtual machine, NS for network segmentation, and Rx for the recommendation sequence.

VM-NS-R1: In environments using virtual switches for network segmentation, it is strongly recommended that distributed virtual switches are used instead of standalone virtual switches for the following reasons: (a) to ensure consistency of configuration across virtualized hosts and reduce chances of configuration errors, and (b) to eliminate constraints on VM migration, since a distributed virtual switch (defined for a particular sensitivity level) by definition spans multiple virtualized hosts.

VM-NS-R2: Isolation of the hypervisor's management network using virtual switches needs special configuration. In addition to dedicated virtual switches, the management traffic pathway should have separate pNICs and separate physical network connections (besides the traffic itself being encrypted). Also, it is preferable that the dedicated virtual switch is a standalone virtual switch (so that it can be configured at the virtualized host level) instead of a distributed virtual switch. This is due to the close dependency between distributed virtual switches and the centralized virtualization management servers. Distributed virtual switches can only be configured using a virtualization management server (requiring high availability for these servers), and in some situations bringing up a virtualization management server may require distributed virtual switch modification.

VM-NS-R3: In all VLAN deployments, the switch (physical switch connecting to virtualized host) port configuration should be VLAN aware – i.e., its configuration should reflect the VLAN profile of the connected virtualized host.

VM-NS-R4: Large data center networks with hundreds of virtualized hosts and thousands of VMs and requiring many segments should deploy overlay-based virtual networking because of scalability (Large Namespace) and virtual/physical network independence. However, it is highly advisable that the overall traffic generated by overlay-based network segmentation technique (e.g., VXLAN network traffic) is isolated on the physical network using a technique such as VLAN in order to maintain segmentation guarantees.

VM-NS-R5: Large overlay-based virtual networking deployments should always include either centralized or federated SDN controllers using standard protocols for configuration of overlay modules in various hypervisor platforms.

3. Network Path Redundancy Configurations for VM Protection

Configuring multiple communication paths for a VM to communicate is essential for ensuring availability, so any network configuration for achieving this can be looked upon as an integral part of network-based protection for VMs. This section discusses options for configuring these paths.

The physical network configuration in a data center is largely unaffected by the presence of virtualized hosts, except for some tasks such as VLAN configuration of ports in the physical switches connecting to the virtualized hosts, and configuration of the associated links as trunk links. The configuration options discussed in this section relating to network path redundancy for VMs are confined to the virtual network inside the virtualized hosts including their pNICs.

The virtual network configuration features provided in most hypervisor offerings involve a combination of load balancing and failover policy options. From a network path redundancy perspective, only failover policy options are of interest.

3.1 NIC Teaming Configuration for Network Path Redundancy

Hypervisor offerings provide a configuration feature called network interface card (NIC) teaming. *NIC teaming* allows administrators to combine multiple pNICs into a NIC team for NIC failover capabilities in a virtualized host.[1] The members of the NIC team are connected to the different uplink ports of the same virtual switch. Failover capability requires at least two pNICs in the NIC team. One of them can be configured as "active" and the other as "standby". If an active pNIC fails or traffic fails to flow through it, the traffic will start flowing (or be routed) through the standby pNIC, thus maintaining continuity of network traffic flow from all VMs connected to that virtual switch. This type of configuration is also called *active-passive NIC bonding*.

Some hypervisor offerings allow NIC teaming functionality to be defined at the VM level. A NIC teaming feature at the VM level enables administrators to create a NIC team using vNICs of a VM. This enables the VM's NICs to perform the same NIC team functionality inside the VM, just like their pNIC counterparts do at the virtualized host level.

3.2 Policy Configuration Options for NIC Teaming

Different hypervisors offer different policy configuration options relating to NIC teaming that may have an impact on failover. These options pertain to different ways in which the NIC team detects NIC/link failure and performs failover. One option detects failures by monitoring electrical signals from the pNIC itself. Another option is to send beacon probes (Ethernet broadcast frames) on a regular basis to detect link failure and configuration problems.

NIC teaming has a significant disadvantage. Because of the limited number of NICs available for each virtualized host, the additional NICs used for virtual traffic path redundancy may reduce the ability to isolate traffic in certain scenarios. For example, in overlay-based network segmentation deployments, data and control plane traffic need to be separated (i.e., carried by different NICs) for network configuration integrity. Hence, such scenarios may call for a tradeoff between two security goals – creating redundant network paths vis-a-vis achieving the necessary logical isolation through network segmentation.

[1] NIC teaming can also help improve virtual network load balancing. This is done to improve performance, not availability, so it is outside the scope of this publication.

3.3 Security Recommendations for Configuring Network Path Redundancy

The following recommendations seek to improve the fault tolerance (redundancy) already provided by NIC teaming. Each recommendation has a unique identifier of format VM-NPR-Rx, where VM stands for virtual machine, NPR for network path redundancy, and Rx for the recommendation sequence.

VM-NPR-R1: It is preferable to use pNICs that use different drivers in the NIC team. The failure of one driver will only affect one member of the NIC team, and traffic will keep flowing through the other members.

VM-NPR-R2: If multiple PCI buses are available in the virtualized host, each pNIC in the NIC team should be placed on a separate PCI bus. This provides fault tolerance against PCI bus failure in the virtualized host.

VM-NPR-R3: The network path redundancy created within the virtual network of the virtualized host should also be extended to the immediate physical network links emanating from the virtualized host. This can be achieved by having the individual members of the NIC team (i.e., the two or more pNICs) connected to different physical switches.

4. VM Protection through Traffic Control Using Firewalls

The primary use of a firewall is for traffic control. In a virtualized infrastructure, traffic control for VM protection is to be exercised for the following two scenarios:

- Traffic flowing between any two virtual network segments (or subnets)
- All traffic flowing into and out of a VM

Scenario 1: Traffic Flowing between Virtual Network Segments or Subnets

There are several use cases where traffic flowing between two VMs (or groups of VMs) need to be controlled, regardless of whether the VMs are resident within the same virtualized host or in different virtualized hosts. The following are examples:

- Applications in an enterprise may be of different sensitivity levels. It may be impractical to fully segregate them by dedicating one or more virtualized hosts to applications of each sensitivity level, so limited segregation may be achieved instead by designating a single application sensitivity level for each VM. Because a given virtualized host may contain VMs of different sensitivity levels, there is a need to control traffic between VMs within the same virtualized host (inter-VM intra-host traffic).
- Most large-scale enterprise applications are designed with three-tier architectures – web server, application, and database. There may be multiple VMs associated with each tier, and for reasons of load balancing and security, VMs hosting applications belonging to a particular tier are generally assigned to the same network segment or subnet although they span multiple virtualized hosts. This type of configuration gives rise to the presence of a web server subnet (segment), database server subnet, etc. However, for any enterprise application to function, the webserver tier needs to talk to the corresponding application tier, which in turn may need to communicate with the database tier. A VM hosting a web server tier and housed in subnet-A needs controlled connectivity to a VM hosting an application tier and housed in subnet-B. Since a subnet itself can span multiple virtualized hosts, it automatically implies that VMs belonging to different application tiers (on a dedicated subnet) may be located in different virtualized hosts and hence the traffic between them needs to be controlled as well (inter-VM inter-host traffic).
- In some enterprises, networks are segmented based on departments in an enterprise (this applies even if the underlying infrastructure is virtualized). In such an environment, the need to exchange data selectively between applications belonging to different departments may require communication between VMs in different segments.

The common requirement in all the use cases discussed above is that all inter-VM traffic must be subjected to policy-based inspection and filtering. Inter-VM traffic is initiated when a VM generates communication packets that are sent through a vNIC of that VM to the port of a virtual switch defined inside the hypervisor kernel. If the target VM resides inside the same virtualized host, these packets are forwarded to another port in the same virtual switch. The target VM (dedicated to it) either may be connected to the same virtual switch, or may connect through another VM that acts as a bridge between the virtual switches of the two communicating VMs. If the target VM resides in another virtualized host, these packets are sent to the uplink ports of that virtual switch to be forwarded to any of the pNICs of that virtualized host. From there, these packets travel through the physical network of the data center and on to the virtualized host where the target VM resides. The packets again travel through the virtual network in that virtualized host to reach the target VM.

Since VMs are the end nodes of a virtual network, the originating and ending networks in any inter-VM communication are virtual networks. A software-based virtual firewall, either functioning in a VM or in the hypervisor kernel, would be a natural mechanism to control inter-VM traffic. However, since connection between any two virtual segments (in different virtualized hosts, at least) goes through a physical network, a physical firewall can also be deployed to control inter-VM traffic between VMs in different virtualized hosts. This was one of the earliest approaches adopted for controlling inter-VM traffic.

A physical firewall configuration to control inter-VM traffic is analyzed for its pros and cons in Section 4.1. A subnet-level (VM-based) virtual firewall approach for controlling inter-VM traffic is discussed in Section 4.2.

Scenario 2: Traffic Flowing Into and Out of a VM

In this scenario, traffic flowing into and out of a particular VM needs to be controlled based on fine-grained policies. To enforce these policies, a mechanism is needed to intercept packets between the vNIC of a VM and the virtual switch within the hypervisor kernel. Such a mechanism is provided by a class of virtual firewalls called *kernel-level virtual firewalls*, *NIC-level firewalls*, or *hypervisor-mode firewalls*. The advantages and disadvantages of this class of virtual firewalls are discussed in Section 4.3.

Firewall Classes

A brief overview of the three classes of firewalls mentioned above (physical firewall, subnet-level virtual firewall, and kernel-based virtual firewall) is given below to facilitate analysis of their advantages and disadvantages.[2]

- **Physical firewalls**: These firewalls can perform their functions in either hardware or software. Their distinguishing feature is that no other software runs on the server where the firewall is installed; in other words, the hardware of the server is dedicated to running only one application, the firewall application.
- **Virtual firewalls**: Virtual firewalls are entirely software-based. The disadvantages and limitation of physical firewalls motivated the development of virtual firewalls. They are distinguished from physical firewalls because they share computing, network, and storage resources with other VMs within the virtualized host where they are installed. The two sub-classes of virtual firewalls are:

  (a) **Subnet-level virtual firewalls**: These run in a dedicated VM, which is usually configured with multiple vNICs. Each vNIC is connected to a different subnet or security zone of the virtual network. Since they communicate with the virtual network only through the vNICs of the VM platform, they are agnostic to the type of virtual network.

  (b) **Kernel-level virtual firewalls**: These firewalls are logically placed between the vNIC of VMs and the virtual switch inside the hypervisor kernel. They function as loadable (hypervisor) kernel modules using the hypervisor's introspection application programming interface (API), which enables them to intercept every packet coming into and out of an individual VM. Subsequent filtering of packets can be performed either in the hypervisor kernel itself or in a dedicated VM. In the latter case, the portion of the firewall functioning as a kernel module performs the function of just intercepting and forwarding the traffic to a VM-based module, and the actual filtering of traffic is done in the VM-based module (just as a

[2] Discussion of application proxies that provide firewall services for applications running a specific protocol (e.g., HTTP for web servers, SMTP for email servers, etc.) is outside the scope of this document.

VM-based subnet-level virtual firewall does). A special class of NIC-level firewall is a distributed firewall (available mostly in solutions providing overlay-based virtual network segmentation), which can be used to filter traffic into and out of any VM located in any virtualized host in the infrastructure.

4.1 Physical Firewalls for VM Protection

In this early scheme, the inter-VM virtual network traffic inside a virtualized host is routed out of that virtual network (often called network in the box) on to the physical network (via the pNICs connected to the uplink ports of the virtual switches to which VM are connected). On this network is installed a firewall with filtering rules pertaining to traffic flowing out of and into each VM on the virtualized host. The VLAN traffic emerging out of the virtualized host is inspected by this firewall and is then either dropped or passed back into the virtual network and on to the target VM.

4.1.1 Advantages

The advantage is leveraging mature, sophisticated firewall rules and other capabilities of physical firewall technologies.

4.1.2 Disadvantages

The use of physical firewalls for inspection and filtering of virtual network traffic carries a number of disadvantages, including the following:

- The performance penalty due to increased latency involved in routing the virtual network traffic to the physical network outside the virtualized host and then back to the virtual network inside the virtualized host. This phenomenon is known as *hairpinning*. The hairpinning route may require multiple hops from the pNIC of the virtualized host to the network location of the firewall appliance, further increasing latency.
- The error-prone manual process involved in maintaining the state information about various VMs as the composition of VMs inside a virtualized host changes due to VM migrations.
- The physical firewall's potential lack of integration with the virtualization management system. This may hamper the automation of provisioning and the update of firewall rules that may be continuously changing because of VM profile changes.
- The overhead and latency in shuttling packets back and forth between the firewall and other service appliances if the firewall is not rich enough to support those services (e.g., application load balancing, policy-based routing).

4.2 Subnet-Level Virtual Firewalls

Virtual firewalls are entirely software-based artifacts, packaged as a virtual security appliance and run on specially prepared (hardened) VMs. The first generation of virtual firewalls operated in bridge mode – that is just like their physical counterpart, they can be placed at a strategic location within the network – in this case the virtual network of a virtualized host. These firewalls can either be stateful or application firewalls. In addition, many of them offer additional features such as network address translation (NAT), Dynamic Host Configuration Protocol (DHCP), and site-to-site Internet Protocol Security (IPsec) virtual private networking (VPN) as well as load balancing for selective protocols such as Transmission Control Protocol (TCP), Hypertext Transfer Protocol (HTTP), and HTTP over Transport Layer Security (HTTPS).

4.2.1 Advantages

There are two notable advantages to subnet-level virtual firewalls:

- They avoid the need to route virtual network traffic from the virtualized host to the physical network and back.
- The effort required to deploy one is as low as deploying any other VM.

4.2.2 Disadvantages

Significant disadvantages to subnet-level virtual firewalls include the following:

- The speed of packet processing is dependent on several factors, such as number of CPU cores allocated to the VM hosting the firewall appliance, the TCP/IP stack of the OS running the appliance, and the switching speed of hypervisor switches.
- In virtualized hosts containing VMs running I/O intensive applications, there could be heavy hypervisor overhead. Even in other cases, since the firewall functions in a VM, it takes away some resources that could otherwise be used for running production applications.
- Since the virtual firewall is itself a VM, the integrity of its operation depends upon its relationship to application VMs. Uncoordinated migration of VMs in the hypervisor could alter this relationship and affect the integrity of the firewall's operation.
- Traffic flowing into and out of all port groups and switches connected with the zones associated with the firewall are redirected to the VM hosting the firewall, resulting in unnecessary traffic (a phenomenon called *traffic trombones*).
- Firewall rules and states associated with a VM do not migrate automatically when a VM is live-migrated to another virtualized host. This may cause the VM to lose the protection provided by the firewall unless the same rules are reconfigured in the environment of the target virtualized host.

4.3 Kernel-Based Virtual Firewalls

Kernel-based virtual firewalls were designed to overcome the limitations of subnet-level virtual firewalls. A kernel-based virtual firewall is packaged as a Loadable Kernel Module (LKM), which means it is installed and run in the hypervisor kernel. Because an LKM must be tightly coupled with the hypervisor executable module, it may not be available on all hypervisor platforms.

4.3.1 Advantages

Kernel-based virtual firewalls have some significant advantages:

- They offer much higher performance than subnet-level virtual firewalls because their packet processing is done using the hardware resources available to the hypervisor kernel instead of the VM-assigned resources (virtual CPUs & virtual memory).
- Since they run as hypervisor kernel modules, their functionality cannot be monitored or altered by a rogue VM with access to the virtual network inside the hypervisor host.
- They have the greatest visibility into the state of the VM, including virtual hardware, memory, storage, and applications, besides the VM's incoming and outgoing network traffic.

- They have direct access to all virtual switches and all the network interfaces of those switches. So the scope of their packet monitoring and filtering functionality not only includes inter-VM traffic but also traffic from the VM to the physical network through the pNICs of the hypervisor host.
- Since these firewalls are hypervisor kernel modules, packet filtering functions operate between the vNICs of each VM and the hypervisor switch. The firewall rules and states are logically attached to the VM interface, so they move with the VM when it migrates to another virtualized host, thus providing continuity of security protection for the migrated VM.

4.3.2 Disadvantages

A notable disadvantage of kernel-based virtual firewalls is that there can be problems integrating them with some virtualization management tools having access to only VMs or virtual networks. Because this class of firewalls runs as a managed kernel process, it is neither a VM-resident program nor a component of the virtual network (such as a virtual switch or a vNIC) of the virtualized host.

4.4 Security Recommendations for Firewall Deployment Architecture

Based on the analysis of the three classes of firewalls (Sections 4.1, 4.2, and 4.3), the following security recommendations for firewall deployment are provided. Each recommendation has a unique identifier of format VM-FW-Rx, where VM stands for virtual machine, FW for firewall, and Rx for the recommendation sequence.

VM-FW-R1: In virtualized environments with VMs running delay-sensitive applications, virtual firewalls should be deployed for traffic flow control instead of physical firewalls, because in the latter case, there is latency involved in routing the virtual network traffic outside the virtualized host and back into the virtual network.

VM-FW-R2: In virtualized environments with VMs running I/O intensive applications, kernel-based virtual firewalls should be deployed instead of subnet-level virtual firewalls, since kernel-based virtual firewalls perform packet processing in the kernel of the hypervisor at native hardware speeds.

VM-FW-R3: For both subnet-level and kernel-based virtual firewalls, it is preferable if the firewall is integrated with a virtualization management platform rather than being accessible only through a standalone console. The former will enable easier provisioning of uniform firewall rules to multiple firewall instances, thus reducing the chances of configuration errors.

VM-FW-R4: For both subnet-level and kernel-based virtual firewalls, it is preferable that the firewall supports rules using higher-level components or abstractions (e.g., security group) in addition to the basic 5-tuple (source/destination IP address, source/destination ports, protocol).

5. VM Traffic Monitoring

Firewalls only ensure that inter-VM traffic conforms to organizational information flow and security rules. However, to identify any malicious or harmful traffic coming into or flowing out of VMs and to generate alerts or take preventive action, it is necessary to set up traffic monitoring capabilities to monitor all incoming/outgoing traffic of a VM. This requires functionality to send copies of those packets to a network monitoring application (also called an analyzer application). This functionality is called *port mirroring*. The purpose of a network monitoring application is to perform security analysis, network diagnostics, and network performance metrics generation. Configuration options are available in hypervisors to turn on port mirroring functionality. Depending upon the hypervisor offering, this configuration option may exist as either a VM-configuration feature or a virtual switch port configuration feature.

5.1 Enabling VM Traffic Monitoring Using VM Network Adapter Configuration

In some hypervisor offerings, the network monitoring application runs as a VM-based application. This VM and its vNIC become the destination VM/vNIC (the *analyzer VM*) to which traffic must be sent for analysis. The VM to have its traffic monitored (the *monitored VM*) becomes the source VM/vNIC. Thus the values "Source" and "Destination" are assigned to the "mirroring mode" configuration parameters of the vNICs of the monitored VM and analyzer VM, respectively.

5.2 Enabling VM Traffic Monitoring Using Virtual Switch Port Configuration

There are two ways that a virtual switch can be configured to enable network monitoring tool visibility into traffic flowing into and out of a particular VM:

- In the earlier versions of a virtual switch, the only configuration option available was to set a particular VM port group into promiscuous mode. This will allow any VMs connected to that port group to have visibility into the traffic going into or coming out of all VMs connected to that port group.
- In the latter versions of a virtual switch, the traffic flowing into and out of the port of a virtual switch (to which the monitored VM is connected) can be forwarded to another specific port. The target or destination port can be another virtual port or an uplink port. This provides flexibility because the network monitoring application can be located either in a VM or in the physical network outside the virtualized host.

5.3 Security Recommendations for VM Traffic Monitoring

Based on the available configuration options in various hypervisor platforms, the following are the recommendations for VM Traffic Monitoring options. Each recommendation has a unique identifier of format VM-TM-Rx, where VM stands for virtual machine, TM stands for traffic monitoring, and Rx for the recommendation sequence.

VM-TM-R1: VM traffic monitoring should be performed for both incoming and outgoing traffic.

VM-TM-R2: If traffic visibility is accomplished by setting the promiscuous mode feature, care should be taken to see that this is activated only for the required VM port group and not for the entire virtual switch.

VM-TM-R3: A port mirroring feature that provides choices in destination ports (either the virtual port or uplink port) facilitates the use of network monitoring tools in the physical network which are generally more robust and feature rich compared to VM-based ones.

6. Summary

With the increasing percentage of virtualized infrastructure in enterprise data centers, the VMs hosting mission-critical applications become a critical resource to be protected. VMs, just like their physical counterparts (i.e., physical servers), can be protected through host-level and network-level security measures. In the case of VMs, since they are end nodes of a virtual network, the virtual network configuration is a critical element in their protection. Four virtual network configuration areas are considered in this publication: network segmentation, network path redundancy, traffic control using firewalls, and VM traffic monitoring. Each area has been analyzed and corresponding security recommendations have been provided.

Appendix A - Acronyms

ACL	Access Control List
CISO	Chief Information Security Officer
CPU	Central Processing Unit
DBMS	Database Management System
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
FISMA	Federal Information Security Modernization Act
GRE	Generic Routing Encapsulation
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Transport Layer Security
IaaS	Infrastructure as a Service
I/O	Input/Output
IPsec	Internet Protocol Security
ITL	Information Technology Laboratory
LKM	Loadable Kernel Module
MAC	Media Access Control
NAT	Network Address Translation
NFS	Network File System
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
OS	Operating System
PCI DSS	Payment Card Industry Data Security Standard
pNIC	Physical Network Interface Card
QoS	Quality of Service
SP	Special Publication
SSH	Secure Shell
STT	Stateless Transport Tunneling
TCP	Transmission Control Protocol
ToR	Top-of-Rack
UDP	User Datagram Protocol
VDI	Virtual Desktop Infrastructure
VLAN	Virtual Local Area Network
VM	Virtual Machine
vNIC	Virtual Network Interface Card
VNID	VXLAN Network Identifier
VPN	Virtual Private Network
VTEP	VXLAN Tunnel Endpoint
VXLAN	Virtual Extended Local Area Network

Appendix B - Bibliography

R. Chandramouli, "Analysis of Network Segmentation Techniques in Cloud Data Centers," *2015 International Conference on Grid & Cloud Computing and Applications (GCA'15)*, Las Vegas, Nevada, United States, July 27-30, 2015, pp. 64-70.
http://worldcomp-proceedings.com/proc/p2015/GCA2935.pdf [accessed 2/1/2016].

R. Chandramouli**, "**Deployment-Driven Security Configuration for Virtual Networks**,"** *Sixth International Conference on Networks & Communications (NETCOM 2014)*, Chennai, India, December 27-28, 2014, pp. 1-13.
http://dx.doi.org/10.5121/csit.2014.41301.

N. Marshall, *Mastering VMware vSphere 6*, Indianapolis: John Wiley & Sons, 2015.

I. Pepelnjak, *Introduction to Virtualized Networking* [Webinar],
http://www.ipspace.net/Introduction_to_Virtualized_Networking [accessed 2/1/2016].

I. Pepelnjak, *Overlay Virtual Networking* [Webinar],
http://www.ipspace.net/Overlay_Virtual_Networking [accessed 2/1/2016].

I. Pepelnjak, *Virtual Firewalls* [Webinar],
http://www.ipspace.net/Virtual_Firewalls [accessed 2/1/2016].

I. Pepelnjak, *VXLAN Technical Deep Dive* [Webinar],
http://www.ipspace.net/VXLAN_Technical_Deep_Dive [accessed 2/1/2016].

D. Shackleford, *Virtualization Security: Protecting Virtualized Environments*, Indianapolis: John Wiley & Sons, 2013.

Z. Shah, *Windows Server 2012 Hyper-V: Deploying the Hyper-V Enterprise Server Virtualization Platform*, Birmingham, United Kingdom: Packt Publishing Ltd, March 2013.

Virtualization Special Interest Group, PCI Security Standards Council, *PCI Data Security Standard (PCI DSS) Version 2.0, Information Supplement: PCI DSS Virtualization Guidelines*, June 2011.
https://www.pcisecuritystandards.org/documents/Virtualization_InfoSupp_v2.pdf [accessed 2/1/2016].