Virtualizing Networking and Security in the Cloud

Debashis Basak, Rohit Toshniwal, Serge Maskalik, Allwyn Sequeira
VMware Inc., Palo Alto, CA

{dbasak, rtoshniwal, smaskalik, asequeira}@vmware.com

ABSTRACT

Virtualization of computer workloads onto powerful x86 multi-core platforms is leading to a massive transformation in the way services are produced by next generation data centers. Simultaneously, cloud computing principles are compelling a re-think in the way enterprises are beginning to consume such services. In this paper, we present the need for network and security (netsec) functions, which are currently realized in hardware appliances, to significantly evolve to keep pace with these new trends, and to provide "disruptively simplified" security that was not earlier possible.

With server consolidation and desktop virtualization, significantly more traffic remains within the data center racks, leading to blind spots for "in network" security appliances. Current netsec devices which are architected based on "scale up" principles cannot keep pace with increased bandwidth driven to the servers, and the ever increasing volume of threats at all layers of the network stack. Also, highly mobile workloads and increasing intelligence in the virtual/hypervisor layer, makes it increasingly hard for static network devices to interlock with dynamic policy changes and on-the-fly re-purposing of resources to serve different workloads, applications, or users.

This paper highlights a new trend in the industry to virtualize *netsec* functions inside security virtual appliances (SVAs), which can then be placed on hosts, and offer distributed security functions for network flows across the cluster. We analyze this trend in detail using the VMware *vShield* product line as an example. The approach replaces single *choke-point* based physical security devices like firewalls, IP address Management (IPAM), flow monitoring, and data leakage prevention *(DLP)* with distributed virtual counterparts running on slices of x86 co-located with compute workloads with ability to tap into traffic going in and out of virtual machines (VMs).

vShield's distributed scale-out architecture means performance can scale up or down linearly as new SVAs are added, while simplifying the lifecycle management of these SVAs including installs, upgrades, ability to debug, and reliability by leveraging underlying virtualization primitives of VM cloning, deploy from template, and VM high availability and fault tolerance. Interactions with features like live migration (*vmotion)* of guest VMs and distributed power management of host servers introduce new aspects of appliance management that was not possible in the physical world. The paper analyzes these aspects of SVA management in depth. Our measurements of the security inspection throughput for given vCPUs and memory indicate it is comparable to those of physical counterparts with the additional flexibility of a scale-out deployment. Further, we demonstrate that with this approach a virtual datacenter (VDC) in the cloud can be deployed in minutes compared to days/weeks with physical datacenters. Finally, we present the additional security inspections that can be performed in the virtual world that were not possible in the physical world. The ability of SVAs to introspect traffic into and out of VMs implies they can perform checks for MAC spoofing, IP spoofing [6], ARP filtering at the source. Furthermore, based on security analysis if a VM is deemed suspect it can be quickly quarantined.

Concepts such as flow introspection, automated insertion of SVAs into flows at VM ingress/egress, distributed scale out architecture across a cluster of hosts, encapsulation of secure VDCs, and programmability of security policies via RESTful interfaces, represent a significant architectural change, with wide applicability in enterprise data centers, and private/public cloud environments.

1. INTRODUCTION

It is becoming a clear trend for organizations to deploy virtual servers instead of physical ones. Recently the number of virtualized servers deployed crossed the number of physical servers.

The process of setting up a new physical data center is onerous and time-consuming. It starts with renting out physical co-location space, to racking and stacking netsec equipment, e.g. firewalls, routers, Dynamic Host Configuration Protocol (DHCP) servers, Virtual Private Networking (VPN) terminators, and configuring these with proper rules to create secure zones, and then finally racking and stacking compute servers. The whole process can take anywhere from days to weeks.

The first simplification is realized by creating an infrastructure of racked and stacked x86 blade servers running hypervisor software to enable virtualization. Next, compute servers are virtualized and hosted on top of this. Such infrastructure offered as a service is referred to as a *cloud*. When offered by a service provider e.g. Amazon EC2, it is referred to as *public cloud*. When owned and operated by an enterprise it is referred to as *private cloud*. Figure 1 shows an example of a VDC deployed in a private or public cloud. This VDC has three zones: DMZ, Web Servers, and App/Database Zone. It requires netsec around these zones like

perimeter firewall, segmentation, NAT, Site to site VPN, Intrusion Prevention System (IPS), and DLP. In this figure the netsec functions are offered using physical appliances in the cloud infrastructure.

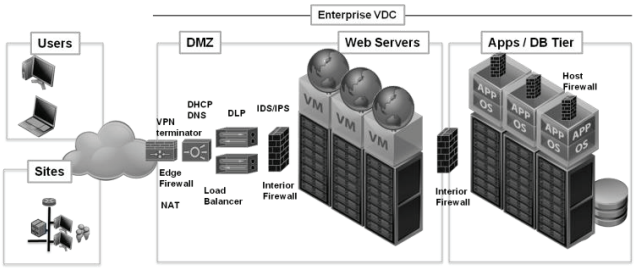


Figure 1: An enterprise data center with virtualized three tier compute workloads hosted in a private or public cloud. Edge netsec functions like perimeter firewall, NAT, DHCP, DNS, site to site VPN, IPS are offered using physical appliances in the cloud infrastructure. Similarly, interior netsec is provided using physical appliances that offer VLANs, application firewalls, and DLP etc.

Next, we discuss the problems with continuing to do netsec using physical appliances. Let's consider the trend in powerful blade servers with tens of blades each with several multi-core x86 processors and one or more 10 Gbps network interfaces. With hypervisor software installed, each blade can host 10-100s of virtual machines. In the meantime the performance of physical network appliances doing network firewall, NAT, DLP, IPS, remain in the 100Mbps range in the low end to 10Gbps range in the high end[3]. One can observe that to keep up with the networking throughput required on a blade, one may need to dedicate a relatively high end appliance for each netsec function for every blade. The resulting proliferation of netsec appliances can be challenging to manage and expensive to maintain. Further, the 10G interface is only for external facing traffic. Given the current numbers of virtual machines that can be hosted on a single blade server, the amount of traffic between machines on the same host server via the backplane can be very high. It might not be architecturally possible to take all this traffic out of the blade, inspect and bring back in. This can lead to blind spots of security that the physical appliances cannot address.

There is another reason that the blind spot problem exists in physical netsec appliances. Traditionally, Ethernet switching and IP routing devices have offered higher throughput than netsec appliances. The network switching and routing decision algorithms are simple enough to be put in silicon for hardware assist. This lends itself to routers and switches with high throughput and port density and therefore the ability to interconnect a large number of servers. On the other hand, netsec appliances e.g. stateful firewall, application firewall, IPS, and DLP, perform filtering and deep packet inspection at higher layer protocols. The complexity of this processing does not lend itself well to customized silicon. These devices tend to leverage x86-based platforms with minimal hardware acceleration resulting in lower throughput compared to switches and routers. Consequently, netsec appliances have to be placed one or more switching or routing hops away from the actual servers to prevent these from becoming a bottleneck to core server to server traffic. This leads to exposure of an attack surface between the netsec point of enforcement and the actual servers thus creating netsec blind spots. For example firewalls cannot filter between servers that share a broadcast domain. Virtualization of servers further moves the access layer into the virtual switching plane thus exacerbating the blind spot problem.

In addition to the blind spot problem, there is also a trend of standard x86 server blades becoming as or more powerful than the netsec appliances. Let us consider the lifecycle of the server blades in comparison to netsec appliances. Server blades can be updated to newer hardware with lesser dependencies than their netsec counterparts. Netsec appliance manufacturers must have a predictable and reliable hardware solution that is well tested. This negatively impacts the ability for a netsec vendor to take advantage of CPU speed curve, while the server blade vendors demonstrate very quick adoption of new and more powerful x86 chipsets. The complex inspection requirements and the inability to adopt fast new x86 architectures contribute to the bottleneck/chokepoint problem that the traditional netsec appliances face in comparison to server blades.

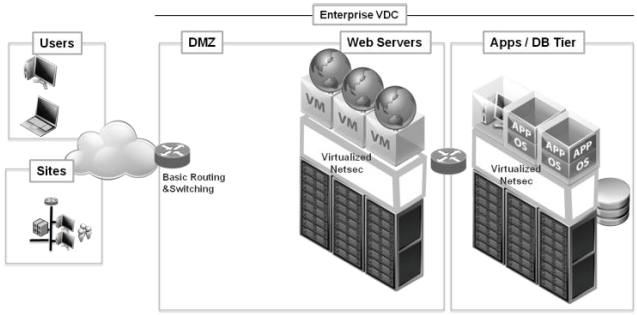


Figure 2: Same VDC as in Figure 1. This time all the netsec functions, except basic routing and switching between host servers, are virtualized and running on slices of x86 co-located with compute workloads. These are depicted as a virtual netsec layer between the hypervisor and the VM workloads.

In this paper we focus on an alternative way to facilitate netsec functions. This involves virtualizing netsec functions and then running these in a distributed manner on slices of x86 co-located with compute workloads. Figure 2 depicts this approach. This time all the netsec functions, except basic routing and switching between host servers, are virtualized and shown as a layer of netsec between the hypervisor and the virtual workloads running on top. We describe a couple of ways to achieve virtualized netsec in the remainder of the paper. Virtualized netsec approach has several advantages: a) it lends to a natural scale-out architecture for netsec with capacity being scaled up or down as needed, b) no separate physical appliances to rack and stack, netsec workloads can be created and deleted in the same flexible manner as virtualized compute workloads, c) ability to transparently ride the x86 performance curve, and d) a point of enforcement that is close

to the virtual machine e.g. firewall functions can be enforced at the network interface of every virtual machine, leading to much more fine-grained control with no blindspots.

In the remainder of the paper we analyze the approach of deploying virtualized netsec using vShield Firewall and vShield Edge as examples. The vShield Firewall is a virtualized firewall and the vShield Edge is a virtualized perimeter appliance that serves as the gateway to a VDC. In Section 2 we discuss the vShield Firewall, its components and how these are deployed on a x86 hardware server running hypervisor (referred to also as *host* in the paper). We also analyze the performance of a virtualized firewall. In Section 3 we discuss vShield Edge and how it leads to efficient deployment of VDCs and analyze its throughput performance. In Section 4 we discuss the additional benefits of virtualized netsec. Section 5 presents concluding remarks.

2. VIRTUAL DISTRIBUTED FIREWALL

In this section we describe the vShield Firewall as an example of a virtualized netsec requiring deployment per hypervisor. As shown in Figure 3 the vShield Firewall on a host comprises of a hypervisor module and a SVA. The vShield SVA is a pre-installed, pre-configured, virtual machine with a hardened operating system specialized for handling firewall operations [2]. The hypervisor module effectively places a network packet filter between the vNIC and virtual switch (vSwitch). It allows the traffic coming in and out of vNICs to be efficiently inspected and if required directed to the vShield Firewall SVA for further processing, depicted using a dashed line. vShield Firewall effectively creates a firewall enforcement presence in front of every vNIC. Every network packet, even those that do not need to leave the host are seen by the vShield Firewall. Thus, the vShield Firewall does not suffer from blind spots that a physical firewall cannot address.

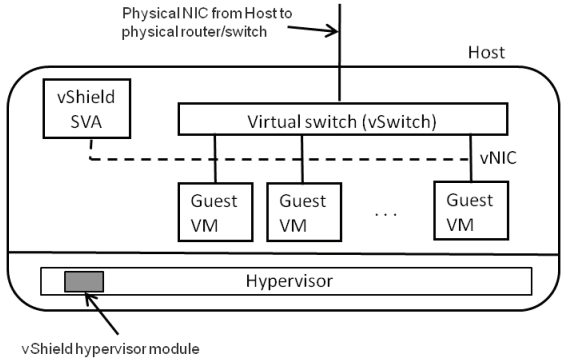


Figure 3: vShield Firewall requires a per host agent. Each agent comprises of a hypervisor module and a SVA to be deployed on the host. The vShield hypervisor module transparently allows traffic going to and from VNICs on guest VMs to be inspected by the vShield Firewall.

Let us consider the deployment of a new physical firewall, it requires moving the appliance to the co-location facility, stacking and racking it, ensuring proper connectivity via cables, accessing the appliance console to configure its IP address so that it can then be accessed from a management console to configure and manage rules. Let us compare this to the deployment of a vShield Firewall on a new host. It is as simple as installing the vShield hypervisor module, then cloning and deploying a SVA, and associating it to the hypervisor module, all achieved programmatically and remotely. Further, creating additional firewall capacity is as simple as deploying a new SVA. It's a distributed scale up model with capacity being added on demand on a given host or on new hosts. Similarly, firewall instances can be simply deleted when a scale down is required. In the remainder of the section we discuss aspects around SVA management that we learned as part of working on vShield.

2.1. Centralized Management

Given the inherent nature of distributed, scale-out model, there is no way an administrator can individually manage firewall rules on every instance on every host. Thus, while there are many policy enforcement points, there has to be a centralized policy configuration point. In this study we used another SVA referred to as vShield Manager that was responsible for management. This included programmatically creating, deploying, upgrading, deleting vShield Firewall appliances. This aspect is similar to the management of physical appliances, except physical appliances cannot be programmatically created or deleted. The manager needs the ability to communicate with the appliances for health monitoring, configuring firewall rules, and receiving alerts, logs, and events. Thus, the manager must be able to reach the appliances via a management network. In the physical world this requires each appliance to have a management interface with an assigned IP address. This IP address along with other network bootstrap information e.g. gateway IP address, subnet mask etc must be entered on to the appliance via its console. This would be cumbersome if the user had to do it for every SVA. Currently, the vShield Manager creates a virtual disk and writes the network bootstrap information for a SVA on it. The disk is then programmatically detached from the manager and attached to the SVA which then mounts it and reads the information to bootstrap itself. In the vShield Edge SVA discussed later, we leveraged the existing host network and piggybacked manager to SVA communication over this network, thus completely obviating the need for a separate vShield management network.

A centralized manager may need to manage hundreds to thousands of SVAs depending on the size of deployment. To ensure the manager does not become a bottleneck, one must partition and contain the domain handled by a single manager. Alternatively, the architecture of the manager must be inherently scalable and distributed. For example, it can have multiple instances that manage different sets of SVAs, with the instances maintaining consistency of configuration and a load balancer distributing configuration requests between them.

2.2. Supporting *vmotion*

In environments that use *vmotion* – live migration of virtual machines around hosts to efficiently share the compute resources

and/or address host failures, one must address the problem of making the distributed netsec function vmotion-safe. vShield is a stateful firewall, which means it keeps track of the state of network connections, such as TCP streams and UDP packets processed by it. The firewall is programmed to distinguish legitimate packets for different types of connections. Only packets matching a known connection state are allowed by the firewall, others are rejected. When a VM vmotions to another host, if the firewall state is not carried along then existing sessions would all get blocked by the destination vShield. By the same token, if the firewall rules are not present on the destination vShield it can lead to a lapse in security. To ensure that live migration of VMs can be supported within a cluster of hosts without loss of security while allowing for operational continuity we need to ensure the following:

1. vShield Firewalls are deployed on all hosts to and from which vmotion is allowed.
2. vShield Manager must dispatch the firewall rules that concern a given VM to all vShield Firewalls where the VM can migrate to. This ensures protection is maintained.
3. Lastly, a vShield Firewall must participate in the *vmotion* protocol which allows it to ensure that the state travel with the virtual machine. This ensures that existing sessions are not blocked.

2.3. Required Constraints on SVAs

In this section we discuss additional considerations on managing SVAs.

Restricted permissions on SVAs: SVAs must be treated as part of infrastructure. Operations like delete and move on these VMs from outside the vShield Manager must be controlled. A casual VM admin should not be allowed permissions to make arbitrary VM operations.

Pinning a SVA to the deployed host: A SVA, e.g. vShield Firewall, is associated with security rules and data path state for VMs on that host. This mandates the vShield SVA itself should never be vmotioned. It should be treated as part of the platform and pinned on the deployed host.

Interaction with Distributed Power Management (DPM): DPM is about reducing energy consumption in the Datacenter. Distributed Power Management allows organizations to cut ongoing power and cooling costs in the datacenter during low utilization time periods. When virtual machines in a cluster of hosts need fewer resources, such as during nights and weekends, DPM consolidates workloads onto fewer servers and powers off the rest to reduce power consumption. When virtual machine resource requirements increase (such as when users log into applications in the morning), DPM brings powered-down hosts back online to ensure service levels are met.

Given the vshield Firewall VM is pinned to the host, it will not vmotion and prevent the host from being powered off. Currently, vShield Manager has to monitor the power off intent on a given host, and then power off the vShield SVA. Similarly, it must power it back on right after the host is brought back up. The ideal solution is to treat the vShield SVA as a special VM and ignore from DPM calculations. DPM should power it off after guest VMs have been vmotioned off. Similarly, DPM should power it back on right after powering on the host but before migrating guest VMs onto it.

High Availability (HA): In the physical world, failover implies deploying redundant appliances. In the virtualized appliance world, there are easier and less expensive answers. Firstly, hardware failure is a non-issue for a vShield Firewall. If the host on which a vShield is running dies then the vShield is no longer available but so are the VMs it was protecting. For protecting against vShield software failures and crashes, one can leverage features like high availability that are already available to VMs. For example, this feature using a heartbeat mechanism has the host monitoring VMs to detect operating system failures. When such a failure is detected it can automatically restart the VM, in this case the vShield SVA.

2.4. Performance of vShield Firewall

Performance of a vShield Firewall was of significant interest in assessing the viability of virtualized netsec approach. In this section we analyzed the performance of a virtualized firewall using a single virtual appliance. The test setup, depicted in Figure 4 and described in Table 1 consisted of a host hardware with VMware ESX hypervisor and vShield Firewall deployed on the host as described earlier in this section. The host also had a guest VM, either a Win2K8 x64 (2GB RAM) or a RHEL5 x64 (2GB RAM), which acted as the server. The benchmark run was *netperf TCP_Stream*.

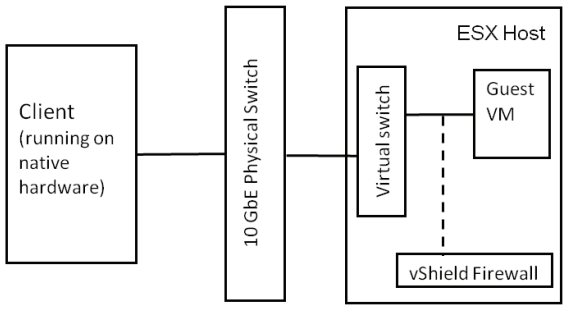


Figure 4: Test setup with a client running on native hardware sending traffic, through a 10Gb physical Ethernet switch and a virtual switch, to a guest VM running on a host. The traffic to/from guest VM and virtual switch is inspected by vShield Firewall.

Results:

Figure 5 summarizes the results observed. The bar graphs show the maximum throughput achieved in Gbps for different configuration parameters of Netperf in TCP_Stream test. The experiment was repeated for three scenarios, depicted using 3 tags : 1) ESX4.1 – traffic sent without vShield Firewall in the path, 2) vShield Firewall with 0 firewall rules, 3) vShield Firewall with 5K rules. The bars on left half are with a Win2k8 server and the

ones on right half are with a RHEL server. Based on the graphs it can be observed that for the Win2k8 server with netperf message sizes greater than or equal to 16K, the native ESX provided up to 9.4 Gbps throughput. With vShield the throughput achieved was less, between 7.5 Gbps and 9.1 Gbps with message sizes of 16K and 128K, respectively. Similar results were observed with the RHEL server.

Table 1: Detailed specification of the test bed.

Host hardware specifications
• Client - 10G (Intel Corporation 82598EB 10 Gigabit AF Dual Port)
• ESX Host
  ○ CPU: Intel Xeon X5560 Nehalem @ 2.8GHz Dual Quad Core (2 socket, 4 cores each)
  ○ DRAM Capacity: 6 GB
  ○ Network Interface – Intel Oplin 10GBE dual-port, Driver: ixgbe 1.3.16.1-LRO with NAPI

Hypervisor specifications running on the hardware • VMware vSphere ESX 4.1
Virtual Firewall (VMware vShield Zones virtual appliance) specifications • 2vCPU • 2 GB DRAM
Traffic Generation Generator: Netperf version 2.4.5 running on a physical RHEL5 server. Destination: Win2K8 virtual machine or RHEL5 x64 (2GB RAM).

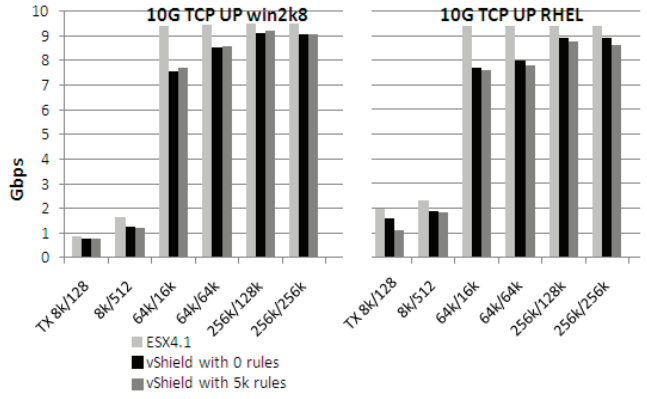


Figure 5: Maximum throughput observed in Gbps for different values of netperf socket/message size in bytes. The experiment was repeated for three scenarios, depicted using 3 tags : 1) ESX4.1 – traffic sent without vShield Firewall in the path, 2) 0-rule – traffic sent through vShield Firewall with no firewall rules, 3) 5K-rule – traffic send through vShield Firewall with 5K rules. The bars on left half are with a Win2k8 server and the ones on right half are with a RHEL server.

A vShield Firewall SVA with 5K rules was able to achieve 9.1 Gbps of throughput with 5 netperf sessions. At this point the appliance was doing close to 370K packets/s. The point to be noted is that such performance can be achieved in a virtual appliance firewall and that it is comparable to the performance of a physical appliance based firewall. These numbers demonstrate the viability of virtualized netsec and dissipate doubts that SVAs can only do few hundred Mbps. Moreover, the added flexibility here is on-demand deployment of netsec in a scale-out fashion. The other benefit is that CPU is not exclusively dedicated to netsec processing. Thus, for a given workload that is more compute than networking intensive, the CPU on the host would be efficiently utilized for computing and not for netsec.

3. DEPLOYING A SECURE VIRTUAL DATA CENTER

In Sec. 1 we discussed the onerous process of setting up a physical data center and deploying physical appliances to ensure appropriate netsec behavior. The entire process is time-consuming and has to be repeated for every customer tenant of the datacenter.

In an evolutionary step towards virtualized compute Cloud environments, the service providers centralized some of the above mentioned netsec functions into multi-tenant purpose built service modules for routers and switches such as firewall, intrusion prevention and server load balancing blades. This approach leads to new interesting challenges: a) service modules were designed for enterprise networks and do not scale to the backplane fabrics of the routers/switches, therefore limiting the throughput and aggregation capabilities of the overall device, b) become a large fault-domain that can affect many tenants when the blade fails or a mis-configuration occurs as compared to a device per customer, c) lack centralized management with self-service capabilities that can be easily delegated to the tenants. Furthermore, since the netsec functions are not performed within the virtualized plane, a very complex interface between the host servers and the physical network needs to be maintained at the Ethernet layer to connect the tenant Local Area Networks (LANs) to their respective contexts on the netsec physical appliances. Typically, this association is achieved using IEEE802.1q VLAN technology. This presents more challenges due to limit on the maximum number of VLANs that can be supported - 4096 and overheads associated with the complexity of VLAN management e.g. VLAN pruning of switch connections, protection against Layer 2 attacks like VLAN hopping, spoofing [6] and spanning-tree exposures.

Compare this to the convenience of having racked and stacked x86 blade servers with automated scripts that have imaged and prepared the servers with hypervisor software and then being able to deploy VDCs programmatically on this platform. In this study we equated the problem of deploying a virtual data center to creating a secure, isolated portgroup[1] on a distributed vSwitch.

[1] A portgroup aggregates multiple ports under a common configuration and provides a point for VM NICs to be connected. There exist technologies to do Layer 2 isolation of one port group

Such a vSwitch must span all the hosts where VMs of the VDC are to be placed. The network interfaces of the virtual machines deployed into the VDC are attached to this portgroup. A vShield Edge SVA provides network edge security and gateway services to the virtual machines in the port group. The vShield Edge connects the isolated, stub network to the shared (uplink) networks and provides common gateway services such as DHCP, VPN, NAT, and Load Balancing. Common deployments of vShield Edge include in the DMZ, VPN Extranets and multi-tenant Cloud environments where the Edge provides perimeter security for the VDCs.

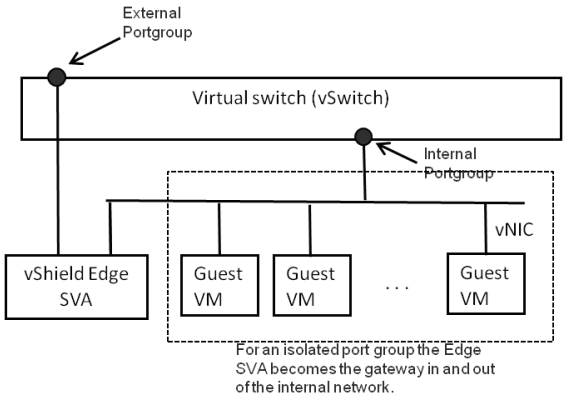


Figure 6: vShield Edge SVA deployed to secure a portgroup that connects the VDC VMs.

Figure 6 depicts a vShield Edge virtual appliance. The SVA has two network interfaces, one external and one internal. The internal interface connects to the secured port group and is the gateway for all protected virtual machines in this port group. The internal interface can have a private IP address block that may overlap with other vShield Edge protected isolated Port Groups. The external interface of the vShield Edge connects to an "uplink" port group which has access to a shared corporate network or a service provide access layer network. It requires at least one external IP address. Multiple external IP addresses can be configured for services like Load Balancer, Site-to-Site VPN or Network Address Translation.

It can be observed that the Edge appliance is deployed one per secure portgroup. The fault-domain with vShield Edge is therefore significantly reduced because unlike in the multi-context switch/router blade approach, failure of a single Edge only affects the tenant being serviced by this Edge. This is akin to what existing datacenter and service provider operators refer to as customer premise equipment (CPE). The vShield Edge is effectively a virtual CPE appliance servicing a single tenant's isolated virtual network.

from another in the virtual layer e.g. using inter-host Ethernet encapsulation. Detailed discussion on portgroup isolation is beyond the scope of this paper.

Deployment of a vShield Edge SVA requires:

1. VM Clone operation to create a new appliance. Connect its external and internal interfaces to the "uplink" and isolated port groups, respectively.
2. Configure the IP address and Subnet Mask for the External interface
3. Configure the IP address and Subnet Mask for the Internal interface, this is used in the DHCP scope and will serve as the default gateway for all virtual machines in this port group.
4. If vmotion and HA are enabled, vShield Edge SVA will be migrated dynamically for most optimal results.

Services available in the vShield Edge at the time of writing include:

- Firewall – supported rules include IP 5-tuple configuration with IP and port ranges for stateful inspection for TCP, UDP and ICMP.
- Network Address Translation – separate controls for Source and Destination IP address and TCP/UDP port translation.
- Dynamic Host Configuration Protocol (DHCP) – configuration of IP pools, gateway, DNS servers and search domains.
- Site-to-Site Virtual Private Network (VPN) – uses IPsec protocol to create a secure tunnel to connect a remote site to the VDC and vice-versa as shown in Figure 7.

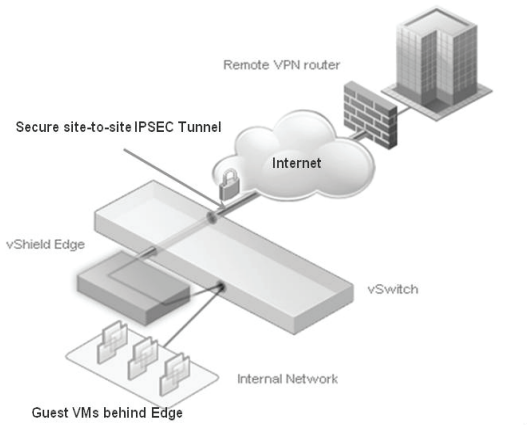


Figure 7: VDC resources being made available to a remote site using site to site VPN to the vShield Edge.

The above services can be configured remotely using web based REST APIs to the vShield Manager. Based on the above discussion, given a ready infrastructure of hosts running hypervisors with virtual switches, a secure VDC can be stood up using the following steps. Each step is a remote CLI call (via REST) over HTTPS to vShield Manager. The Manager further forwards the commands to the appropriate Edge SVA for enforcement. We ran this script to create 1000 VDCs. Table 2 shows the time that each step below required to complete.

- Step1: Create an isolated internal portgroup on a Distributed vSwitch and clone and deploy a vShield Edge.
- Step2: Configure Edge Services:
  - Configure a DHCP pool
  - Configure NAT rules. In this study we configured 100 NAT rules (50 SNAT rules and 50 DNAT rules).
  - Add 100 firewall rules
  - Add 1 Site-to-Site VPN tunnel
- Step3: Add a new guest Windows XP guest VM into the VDC and attach it to the internal portgroup network. Our study focused on bringing up a secure VDC and not on importing of VM workloads into the secure VDC. A single VM was added as a sample to test and validate that the services were configured correctly.

Table 2: Minimum and maximum times to deploy 1000 VDCs, with 10 VDCs being deployed in parallel.

Steps	Minimum Time (secs)	Maximum Time (secs)	Average Time (secs)
Edge SVA Deploy	81	235	110
Configure DHCP pool	6	49	12
Configure SNAT rules	7	86	16
Configure DNAT rules	7	94	15
Configure Firewall rules	7	89	27
Configure VPN tunnel	6	90	15
Add a new guest VM to VDC	6	90	17
Total time to deploy a secured VDC	161	363	212

The above times were obtained without making any obvious optimizations e.g. bulk configure all edge services at same time. Even without that a few key observations to be noted are:

- The time to bring up a VDC is in minutes, not hours or weeks as used to be the case with bringing up a physical DC.
- The average rate of bringing up VDCs is 10 per 212 secs, or 3 VDCs/minute, or 180 VDCs/hr. We believe the parallelism can be increased with further optimizations.

3.1. Additional Security with vShield Edge

The vShield Edge serves as delimiter of a tenant and could implement the same set of policies a customer would normally have in place in a physical datacenter/rage/rack. Filters on the Edge can protect against attacks on the service provider (SP) switching and routing infrastructure initiated by malicious tenant VMs. Below are examples of such attacks and corresponding filters.

Bogon Filtering: RFC 2827 [6] recommends that service providers police their customers' traffic by dropping traffic entering their networks that is coming from a source address not legitimately in use by the customer network e.g. as specified in RFC1918 [5] and RFC3330 [7].

Directed broadcast: Host bits set to ones should be dropped at the external interface to protect against Smurf attacks [8, 9].

IP Source Routing option should be disallowed**.** Source Routing is a technique whereby the sender of a packet can specify the route that a packet should take through the network. It can be used in a number of ways for hacking purposes e.g. used with trace route to map the routing points in the SP network or to reach a specific SP machine.

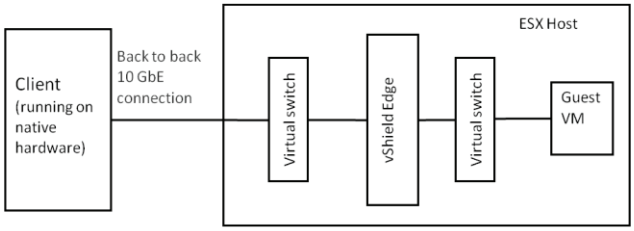
ICMP filtering: Only allow ECHO, ECHO reply, TTL Expired (for traces to work) and block everything else.

"Half-open" connections: Limiting this number in a given time frame to avoid resource Denial of Service (DoS).

Ping Floods: Rate-limit the ICMP data rates to avoid ping floods.

3.2. Performance of a vShield Edge SVA

In this section we analyze the performance of a vShield Edge virtual appliance. The test setup is depicted and described in Figure 8. It consisted of a host running hypervisor and vShield Edge deployed on the host as described earlier. The host also had 1 RHEL Server Guest VM running on it. Traffic was generated by netperf running on a RHEL client on a physical server to the RHEL Server Guest VM. The traffic was inspected by the vShield Edge. Further details on the specifications of the hardware, hypervisor, vShield Edge virtual appliance, and traffic generation are provided below.



Host hardware specifications
- CPU: 2 x Intel (R) Xeon (R) CPU E5520 @ 2.27GHz
- DRAM Capacity: 12 GB

Hypervisor specifications running on the hardware
- VMware vSphere ESX 4.1

vShield Edge virtual appliance specifications
- 1vCPU
- 256 MB RAM

Traffic Generation

Generator: Netperf version 2.4.5 running on a physical RHEL5 server, NetServer and other in-house benchmarks for establishing multiple connections.

Figure 8: Test setup for analyzing performance of vShield Edge.

Results: Table 3 summarizes the Firewall, NAT, and VPN throughput results. Firewall and NAT throughput of 4 Gbps with UDP and 3 Gbps with TCP with 64K concurrent sessions was achieved. With IPSEC VPN a throughput of 220 Mbps with AES and 112 Mbps with 3DES was observed. It is to be noted that in these tests the Edge SVA used only had a single vCPU. These numbers are in line with what is achievable with low end single CPU based x86 physical appliances today.

Table 3: Summary of Results

FireWall and NAT:

• Throughput :
  ○ UDP: With 1, 400, and 2000 rules a single UDP session (netperf) can push 4Gbps traffic.
  ○ TCP: 3Gbps using 4 sessions.
• Maximum number of concurrent sessions supported 64K
• Maximum new session rate:
  ○ Under no background load - 9K sessions/sec
  ○ Under 50% load (by background traffic) - 5.5K sessions/sec

VPN:

• Throughput :
  ○ 220 Mbps with AES encryption
  ○ 112 Mbps with 3DES encryption
Note: This is purely CPU bound and in sync with the throughput predicted based on the CPU available and the processing associated with different encryption algorithms.

4. BETTER THAN PHYSICAL SECURITY

In this section we highlight some important benefits of virtualized netsec that was not possible in the physical world. Earlier we established that a virtualized netsec deployment like vShield Firewall effectively creates a firewall enforcement presence in front of every vNIC. Every network packet, even those that do not need to leave the host are seen by the vShield Firewall. Thus, the vShield Firewall does not suffer from blind spots that a physical firewall cannot address. The vNIC level firewall allows us to achieve additional security policies with ease and create a secure Ethernet transport environment.

Additionally, virtualization offers unique vantage point to centrally collect authoritative knowledge of virtual machine information like assignment of MAC, IP addresses. Introspection techniques can be leveraged to discover applications and services running within a virtual machine. This would allow for very precise spoofing controls and granular firewall rules. In the physical network devices, such details of endpoint computers are not known. These rely on models based on techniques like snooping DHCP responses to populate a list of allocated addresses to computers or network port scans to determine the applications running within the endpoint. These can be prone to error and or not offer fine grained enforcement control. For example, Ethernet port security techniques do not work in the virtualized compute environments because of large number of MAC addresses seen on the same port as a hardware server can host 10-100s of virtual machines and vmotion operations move the virtual Ethernet adapters between physical switch ports. This makes it impossible to lock in a MAC address filter on a physical switch port. The virtualized netsec plane allows for authoritative assignment and validation of such parameters and therefore not prone to such problems. While detailed discussion on this topic is beyond the scope of this paper and will be addressed in future work, we discuss a few interesting examples below.

Prevent MAC and IP spoofing from VMs: We created an IP address management interface on the vShield Manager. This interface is used to authoritatively maintain a list of MAC and IP address for every vNIC. For convenience, optionally the list can be seeded with MAC and IP of a vNIC the first time such information is seen by the virtualized management layer. This can be manually overridden. The list is pushed down to the vShield Firewalls which inspect every packet originating out of a vNIC for the prescribed MAC and IP. If it does not match the packet is simply dropped. This prevents malicious VMs from spoofing other MACs and IPs. It also prevents related attacks like CAM table overflow on the physical router interconnecting the hosts.

Prevent DHCP starvation: The vShield Firewall can perform ARP rate limiting and prevent DoS attacks against the DHCP server.

Securing physical routers and switches from malicious VMs: We already discussed the prevention of CAM table overflow and ARP poisoning. Another concern is malicious VMs generating spoofed network control packets. vShield Firewall can be configured to drop any network control packets e.g. Cisco Discovery Protocol, Spanning Tree BPDUs, VRRP/HSRP multicast frames, dynamic routing interior routing protocols like OSPF, EIGRP, IS-IS from VMs [4]. This is of particular value to service providers who offer VM hosting services. Such policies can protect the service provider infrastructure from spurious packets generated from hosted VMs.

Isolating suspicious VMs: Given a security alert for a VM, e.g. an IPS trigger on a VM, the vShield Firewall can be programmed dynamically to quarantine the VM using a firewall rule.

5. CONCLUSIONS AND FUTURE WORK

In this paper we analyzed the trend in the industry to virtualize *netsec* functions inside security virtual appliances (SVAs) using vShield Firewall and Edge. We demonstrated that vShield's distributed scale-out architecture means performance can scale up or down linearly as new SVAs are added, while simplifying the lifecycle management of these SVAs including installs, upgrades, delete, and high-availability by leveraging underlying virtualization primitives of VM cloning, deploy from template, and VM high availability. We also demonstrated that the vShield Firewall and Edge throughput is comparable to physical appliances, with the added flexibility of an on demand scale-up, and use of CPU resources only on an as needed basis. Further, we demonstrated that with virtualized netsec, a virtual datacenter in

the cloud can be deployed in minutes compared to days/weeks with physical datacenters. Finally, we presented the additional security inspections that can be performed in the virtual world that were not possible in the physical world.

While this study focused on demonstrating the efficiencies of virtualized netsec, we are also investigating advantages of moving guest-based netsec functions e.g. guest-firewall and anti-virus outside the VM into SVAs. Given it is a common practice to perform operations like snapshot, pause and rollback on virtual machines, agent-based solutions can pose a problem by also rolling back to an out-dated state. Additionally, we are addressing trusted platform initiatives and investigating effective ways to validate SVAs to include them as part of a trusted platform.

6. REFERENCES

[1] Dubrawsky, I., “Safe Layer 2 Security in-depth— version 2”, (http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/sfblu_wp.pdf)

[2] “Best Practices for Building Virtual Appliances”, (http://www.vmware.com/go/vam_va_fd_bestpractices&rct=j&q=building%20service%20virtual%20appliance&ei=eLSeTLTGAoWqsAOzzszVAQ&usg=AFQjCNHue62QtQavkxxcqJ79aTMNSTPOOg).

[3] "Cisco ASA Model Comparison Page" as on 2010-09-30.

[4] SANS Institute Recommended Firewall Checklist www.sans.org/score/checklists/FirewallChecklist.pdf.

[5] Rekhter, Y., Moskowitz, R., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", IETF RFC 1918, February 1996.

[6] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", IETF BCP 38, RFC 2827, May 2000.

[7] IANA, "Special-Use IPv4 Addresses", IETF RFC 3330, September 2002.

[8] D. Senie, "Changing the Default for Directed Broadcasts in Routers", IETF RFC 2644, August 1999.

[9] "CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks" (http://www.cert.org/advisories/CA-1998-01.html).