LEARNING MADE EASY

Internal Firewalls

for dummies®
A Wiley Brand

Secure east-west network traffic

Prevent attackers' lateral movement

Deploy distributed internal firewalls

R. Dube

VMware Special Edition

Internal Firewalls

VMware Special Edition

by R. Dube

for dummies®
A Wiley Brand

Internal Firewalls For Dummies®, VMware Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-77296-5 (pbk); 978-1-119-77298-9 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. Some of the people who helped bring this book to market include the following:

Development Editor: Stephanie Diamond

Project Manager: Martin V. Minner

Senior Managing Editor: Rev Mengle

Acquisitions Editor: Ashley Coffey

Business Development Representative: Karen Hattan

Production Editor: Tamilmani Varadharaj

VMware Contributors: Stijn Vanveerdeghem, David Meier, Todd Nugent, Baruch Deutsch, Dhruv Jain, Daniel Peralez

Table of Contents

Introduction

Most medium and large organizations are digital organizations. Many of these organizations have sophisticated information technology (IT) infrastructure that has been added to over the years. The bulk of this infrastructure uses Transmission Control Protocol/Internet Protocol (TCP/IP) for its networking stack and some version of Microsoft Windows or Unix/Linux as the endpoint operating system. Unfortunately, TCP/IP was designed for openness, not security. Although some operating systems are more secure than others, most in use today were not designed for security from the ground up. Finally, applications running on operating systems have become byzantine, sometimes with millions of lines of code. As a result, security holes in the IT infrastructure abound. Cybercriminals know this and are always on the lookout for their next easy target.

Back when IT infrastructure was smaller and less complex, and cybercrime was less prevalent, it was possible to insert a small number of edge firewalls between the outside world and the organization to protect the organizations' infrastructure from external attackers — up to a point. Even then, the edge firewall would not protect an organization against a malicious insider.

These days, cybercrime is big business. Many organizations are too big and juicy a target to attackers to rely solely on edge firewalls for network security. Once attackers get past the edge firewall, they can move laterally in the organization's IT infrastructure with a great deal of freedom. Attackers use this freedom to move from low-value, lightly-defended assets that they have compromised to high-value assets such as databases of personal information or intellectual property stores.

Defenders — security teams — at these organizations need to prevent the lateral movement of attackers. They need to think about compartmentalizing their network to limit damage from individual intrusions. They need to think about obscuring one part of their infrastructure from another. In short, defenders need to think about *internal firewalls.*

About This Book

Welcome to *Internal Firewalls for Dummies*, VMware Special Edition. This book discusses how internal firewalls can help your organization secure east-west network traffic and prevent attackers' lateral movement. It shows how distributed internal firewalls combine the best of hardware-based enterprise edge firewalls and software-based micro-segmentation solutions. It also describes the typical approach taken by organizations to successfully deploy distributed internal firewalls.

Icons Used in This Book

REMEMBER

The Remember icon highlights information that will help you develop a solid understanding of internal firewalls.

TIP

The Tip icon calls out information that will help you make better decisions regarding internal firewalls.

WARNING

The Warning icon denotes items that help you avoid potentially costly mistakes.

IN THIS CHAPTER

- Appreciating the limitations of traditional firewall designs and micro-segmentation solutions
- Summarizing how distributed internal firewalls prevent lateral movement
- Getting started with distributed internal firewalls

Chapter 1
Understanding Why Internal Firewalls Matter

No organization wants to see its name in the same headline as the words "massive data breach." Still, organizations of all sizes make the news as cybercriminals breach their defenses to exfiltrate sensitive data. Traditional security architectures clearly aren't enough to thwart successful attacks. What then, is a security team to do?

This chapter looks at the fact that security teams need to assume that their perimeter defenses (including their edge firewalls) will eventually be breached. They need to think seriously about preventing the lateral movement of attackers inside the organizations' networks.

Preventing Lateral Movement

The network perimeter was once well-defined and well-defended but has become highly permeable because of mobile and remote end-users, personal devices on organizations' networks, and workloads (components of applications) in the public cloud. Even the data center, which once only hosted specialized equipment

and applications, now hosts end-users through virtual desktop infrastructure technology. Thus, mission-critical applications are only a short hop away from attackers' end-user soft targets.

WARNING

Once breached, perimeter defenses can't stop an external attacker from moving laterally inside the organizational network to reach and exfiltrate data records. Often, attackers dwell on the network for weeks or months. Making matters worse, attacks involving insiders, who are already within the perimeter, account for a growing percentage of breaches. The insiders may not even be willing participants in the breach — their credentials or end-user devices may have been compromised by an external attacker.

Instead of relying exclusively on perimeter security, organizations must focus on detecting and blocking malicious east-west (internal) network traffic as a core component of their information technology (IT) security strategy. This requires an internal firewalling approach specifically designed to protect large volumes of east-west data center traffic without sacrificing security functionality, network performance, or manageability. Such an approach would not only improve the organization's security posture, but it would also lower the total cost of ownership of the organization's network defenses.

Appreciating the Limitations of Enterprise Edge Firewall Designs

Today, most internal firewalls descend from enterprise edge firewalls designed to secure limited amounts of traffic moving in and out of organizations (north-south traffic). However, in modern data centers, the volume of east-west traffic is much higher than that of north-south traffic. Further, security teams want to specify fine-grained policies at the level of workloads in the data center. There is no easy way to define such policies with traditional firewalls.

REMEMBER

In the context of modern, distributed applications constructed from dynamic workloads, securing all or even most east-west traffic has long been viewed as too complicated, expensive, and time-consuming for brownfield — and even greenfield — data centers. This perception is undoubtedly accurate for those

organizations that attempt to secure east-west traffic by employing traditional, appliance-based firewalls as internal firewalls.

Traditional enterprise edge firewall designs are a poor fit for internal firewalling — the traffic volume is too great, and the policy granularity required is too fine for enterprise edge firewalls to prevent lateral movement in the data center.

Recognizing Why You Need More than Micro-Segmentation

Micro-segmentation is a method of segmenting the network at a fine-grained level — finer than the coarse-grained segmentation of enterprise edge firewalls. Micro-segmentation came along in 2013 to solve the traffic scale and granularity requirements of securing modern applications. However, micro-segmentation solutions introduced some problems of their own.

REMEMBER

Most micro-segmentation solutions do not implement the full suite of security functionality provided by enterprise edge firewalls. These solutions are orchestrators that use the firewall embedded in the operating systems hosting the workloads.

Micro-segmentation orchestrators are restricted by the capabilities of the stock operating system firewalls. In particular, the orchestrators cannot understand or enforce policies based on users or applications and do not include any threat control mechanisms such as intrusion detection/prevention systems (IDS/IPS), network traffic analysis/network detection and response (NTA/NDR), or sandboxing.

Understanding Distributed Internal Firewalls

Distributed internal firewalls borrow distributed enforcement from micro-segmentation solutions to handle east-west traffic scale and granularity requirements. Simultaneously, they retain the enterprise edge firewall's ability to create and enforce security policies based on users and applications and include threat controls such as IDS/IPS, NTA/NDR, and sandboxing.

REMEMBER

That is, distributed internal firewalls combine the desirable capabilities of traditional enterprise edge firewalls and micro-segmentation solutions to create an innovative firewall architecture.

Extending Distributed Internal Firewalls

Distributed internal firewalls don't just work with virtualized data centers, where workloads are hosted on a hypervisor (virtualization software) on the physical server. They also work with physical servers (without hypervisors), containers, and the public cloud in a similar way to virtualized servers.

REMEMBER

Thus, distributed internal firewalls can enforce a uniform set of policies across workloads, irrespective of the underlying infrastructure (on-premises or public cloud) or the workload type (virtual machine, physical server, or container).

Getting Started With Distributed Internal Firewalling

TIP

Many organizations choose to start with a macro-segmentation (coarse-grained segmentation) deployment, where the distributed internal firewall replicates the organizations' existing network segmentation policies. Such an approach not only builds the security team's confidence in the new approach, but it also gives them the flexibility to quickly and easily create new network segments.

VMware's NSX Service-defined Firewall implements a distributed internal firewall, as shown in Figure 1-1. The Service-defined Firewall is deployed by thousands of organizations. Through these deployments, VMware has learned that getting started with distributed internal firewalls is straightforward.

Over time, the security team extends the deployment to business-critical applications and, ultimately, to all data center applications. They also add IDS/IPS to meet compliance requirements, creating a second defensive layer for the more sensitive applications.

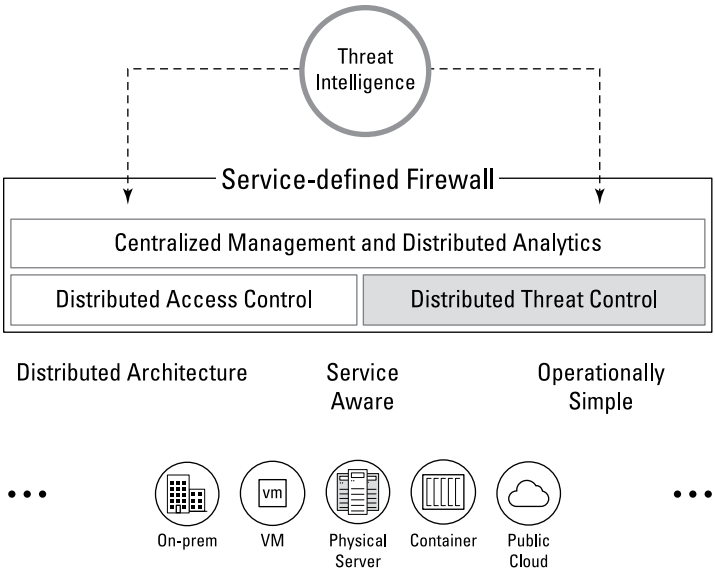

FIGURE 1-1: NSX Service-defined Firewall components.

IN THIS CHAPTER

- Understanding different types of traffic
- Looking at the types of firewalls that protect each traffic type
- Summarizing firewall use cases relevant to east-west traffic

Chapter 2
Recognizing Traffic and Firewall Types

This chapter begins by reviewing the layout of the different traffic types in organizations' networks and the firewall types that have evolved to protect these traffic types. After defining the traffic and corresponding firewall types, the chapter shows you some vital firewall use cases.

Looking at Network Traffic Taxonomy

Network traffic in organizations is defined by the types of entities at the two ends of the communication, as shown in Figure 2-1.

Considering north-south traffic

North-south traffic is network traffic that moves in and out of an organization's network — for example, to and from the Internet. In this case, one end of the communication is inside the network and the other outside. North-south traffic typically represents a small fraction of the overall traffic on a large organization's network.

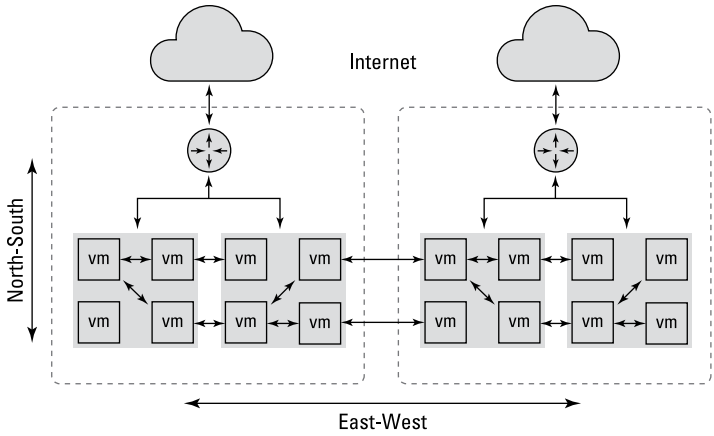

FIGURE 2-1: North-south and east-west (internal) traffic.

Defining internal traffic

Internal traffic is network traffic that moves within an organization's network. Both ends of the communication are within the organization's network.

Three types of internal traffic exist:

- **East-west traffic:** This is traffic between workloads (see the sidebar, "Applications and Workloads") that are constituents of an application, as shown in Figure 2-2.

  East-west traffic also includes inter-application traffic, as well as traffic between applications and shared information technology (IT) services. The workloads, applications and shared services may reside in the organization's data center or public cloud. As a result, this traffic type includes intra-data center, inter-data center, data center to public cloud, and public cloud to public cloud network traffic.
- **User-to-user traffic:** This is direct network traffic between two users. Such traffic is often forbidden in organizations, and its presence usually indicates that something is wrong in the network.
- **User-to-workload traffic:** This type of traffic results from a user in an organization accessing an internal application.

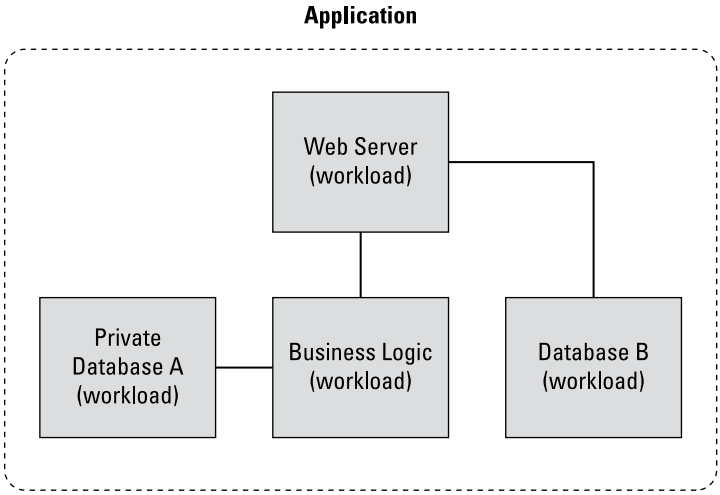

FIGURE 2-2: Applications and workloads.

APPLICATIONS AND WORKLOADS

Different parts of the information technology (IT) industry use the terms *workload* and *application* differently. In data centers, an application is a collection of workloads.

In turn, a workload provides a specific functionality (for example, database functionality) to the application. Three types of workloads exist: virtual machines, physical servers, and containers. In the case of virtual machines and physical servers, the term *workload* includes the operating system on which the particular functionality runs.

This book uses the phrase "hosted on a virtual machine" to mean a workload type where the workload-specific functionality runs in a virtual machine. Similarly, "hosted on a physical server" implies a workload type where the workload-specific functionality runs on a physical server without a hypervisor or container. Finally, "hosted on a container" implies a workload type where the workload-specific functionality is containerized (regardless of whether the container sits on a physical or virtual machine).

As you can imagine, in modern data centers, east-west traffic volume far surpasses user-to-workload traffic.


TIP

This book uses *east-west traffic* to mean all internal traffic. This term clearly contrasts internal traffic with north-south traffic.

Reviewing Network Firewall Taxonomy

Network firewalls inspect network traffic, permitting or blocking traffic flows based on configured security policies. Today, most network firewalls are *stateful* — they constantly track information (they "keep state") on traffic flows to determine whether the traffic flows remain legitimate.


REMEMBER

Network firewall types evolved as network traffic evolved. One class of firewalls, *edge firewalls*, emerged to inspect north-south traffic. Another class, *internal firewalls*, emerged to protect east-west traffic, as shown in Figure 2-3.

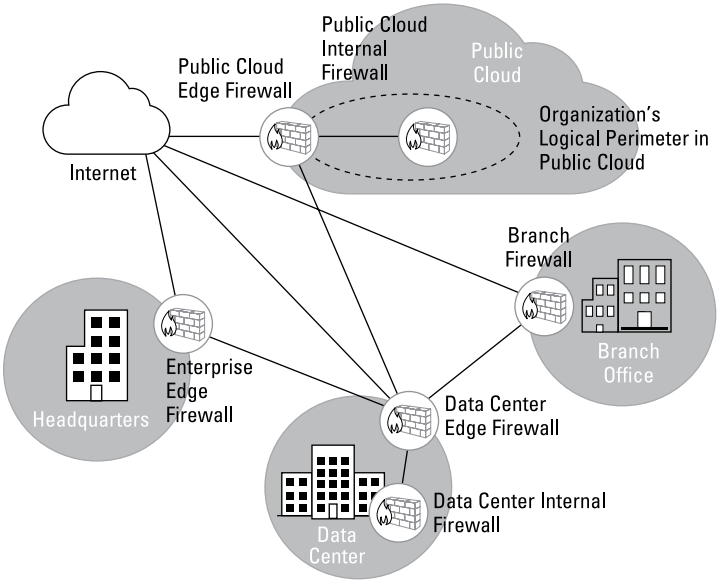

FIGURE 2-3: Firewall types.

Dealing with edge firewalls

Edge firewalls are identified by their location between an organization's internal network and the outside, usually the Internet. An edge firewall is the last firewall between a traffic flow originating inside an organization's network and the Internet (or the first firewall encountered by a traffic flow from the Internet into an organization).

You may also hear the term *perimeter firewall* to refer to edge firewalls. For consistency, this book exclusively uses *edge firewall.*

Four types of edge firewalls exist:

- **Enterprise edge**: The enterprise edge firewall sits between an organization's network at the headquarters or a large campus and the Internet.
- **Branch:** The branch firewall sits between an organization's remote (usually small) office and the data center. It also splits Internet-bound traffic directly to the Internet when such traffic is permitted.
- **Data center edge:** The data center edge firewall sits between an organization's data center network and the Internet.
- **Public cloud edge**: The public cloud edge firewall sits between an organization's public cloud network, even though it is usually a logical network rented from a public cloud provider, and the Internet.

An organization can have multiple campuses, data centers, and public cloud deployments. Such an organization will typically have numerous installations of enterprise edge, data center edge, and public cloud edge firewalls.

Viewing internal firewalls

Internal firewalls are also identified by their location — they are inside an organization's network. Internal firewalls process east-west traffic internal to an organization.

Two types of internal firewalls exist:

- **Data center internal:** The data center internal firewall inspects east-west traffic within a data center, between data centers of the same organization, and between the data center and the organization's public cloud deployments. This type of firewall can also process user-to-workload traffic depending on an organization's network architecture.
- **Public cloud internal:** The public cloud internal firewall is deployed in the public cloud and inspects east-west traffic between the data center and the public cloud and between multiple public cloud deployments.

Collapsing multiple firewall types into one

The firewall taxonomy applies well to large organizations. However, smaller organizations often combine firewall types to simplify their network and security design. For example, medium-sized organizations may combine data center edge and data center internal firewalls into a single data center firewall.

Similarly, small organizations may combine enterprise edge, data center, and data center internal firewalls into a single firewall. Finally, some public cloud deployments may combine the public cloud edge and public cloud internal firewalls.

The branch firewall often acts as both an edge and an internal firewall. Commonly, the branch firewall processes north-south traffic between the branch and the Internet and east-west traffic between the branch and the organization's data center.

Not all branch firewalls are configured to send or receive traffic to the Internet. Some organizations choose to "backhaul" all traffic from the branch to the data center. Traffic destined for the Internet is then sent out through a data center edge firewall. In this case, the branch firewall acts as an internal firewall.

Modern branch networking enables a third mode of branch firewalling: The branch networking appliance does basic firewalling for in-branch traffic. Other traffic is sent to a service provider's local point-of-presence that implements additional security functionality and determines whether the traffic should be sent to the data center, a business partner, or the Internet.

Utilizing other firewalls

This book focuses on network firewalls used in information technology (IT) environments. These are firewalls that, at a minimum, can inspect Layers 2 through 4 (data link, network, and transport layer headers) of data packets in a traffic flow. Frequently, network firewalls can inspect traffic at Layer 7 (the application layer) as well.


REMEMBER

Application firewalls, such as web application firewalls (WAFs), operate at Layer 7 (the application layer). Application layer firewalls may also inspect traffic at Layers 2 through 4, but because their primary focus is at Layer 7, they are not covered in this book.

Some specialized network firewalls run on networks such as operational technology (OT) networks with specialized requirements. Firewalls on OT networks are physically rugged (some are resistant to water, extreme temperatures, and vibrations) and can understand OT network protocols such as those used by industrial control systems. These firewalls are also outside the scope of this book.

Preventing Lateral Movement on East-West Traffic

The primary function of internal firewalls is to prevent the lateral movement of attackers.

The attackers are often outsiders who have breached or circumvented an organization's perimeter defenses and established a foothold on computing equipment inside the organization's network. Sometimes the attacker is a malicious insider. At other times an unsuspecting insider has been compromised and is being used by external attackers to infiltrate the organization.

Internal firewall use cases are recipes that the security industry has created to systematically prevent lateral movement. These recipes impede both external and internal attackers.

Segmenting networks quickly

Network segmentation is a network security hygiene concept that seeks to chop the network into "segments" and restrict traffic in and out of that segment (see the sidebar, "Network Segmentation, Macro-Segmentation and Micro-Segmentation"). Many organizations struggle to apply network segmentation to their internal network. Failure to segment the internal network enables the unfettered lateral movement of attackers once they have circumvented an organization's perimeter.

WARNING

The ideal internal firewall would enable security teams to quickly set up network segments and modify them as the organization's IT infrastructure evolves. Unfortunately, most internal firewalls are not flexible enough to allow rapid changes to network segments.

Meeting compliance requirements

Many organizations operate in regulated industries and need to comply with specific regulations to safeguard data. Example regulations include:

- Health Insurance Portability and Accountability Act (HIPAA)
- Payment Card Industry Data Security Standard (PCI-DSS)
- Sarbanes-Oxley Act (SOX)
- Gramm-Leach-Bliley Act (GLBA)

Some organizations have internal rules that the network and security implementation must conform to.

Most of these rules and regulations are designed to encourage segmentation to prevent attackers' lateral movement. Typically, security teams configure their internal firewalls to police east-west traffic to meet the compliance requirements.

Achieving a Zero Trust Network Architecture

The traditional "castle-and-moat" perimeter security model has proven inadequate for protecting modern IT environments.

NETWORK SEGMENTATION, MACRO-SEGMENTATION, AND MICRO-SEGMENTATION

Network segmentation is a network security technique that divides a network into smaller, distinct sub-networks and enables security teams to control each sub-network's security policies.

Network segmentation limits the local traffic belonging to a sub-network, just to that sub-network. As a result, attackers outside the sub-network cannot surveil the local assets in the sub-network. If attackers manage to break into a workload on a segmented sub-network, the attackers' visibility of attackable assets is limited to the other workloads in that sub-network. Thus, network segmentation impedes the lateral movement of attackers.

Network segmentation is a particular type of coarse-grained segmentation that uses the structure of network addresses to define segments. Coarse-grained segmentation is referred to as *macro-segmentation* (to contrast it with micro-segmentation).

The concept of segmentation can be applied at a finer granularity than sub-networks in network segmentation. A properly constructed sub-network may still consist of tens or hundreds of physical servers and workloads. When one segments the network at the level of workloads that make up an application, the granular segments are called *micro-segments.* The process of creating such segments is *micro-segmentation.*

Security policies control the communication between the micro-segments and the rest of an organization's network. Additional policies may further control the interaction between workloads within the micro-segment. Micro-segmentation is desirable because it restricts the attackers' visibility and lateral movement (should the attackers succeed in accessing a workload) much more than network segmentation.

The "castle-and-moat" perimeter security model inspired by medieval castles established a perimeter using firewalls and other network security devices (the moat) to protect the assets inside (the castle). These days, perimeter network security is not a strong enough deterrent to attackers, especially if organizations' internal network is flat (see the sidebar, "Flat Networks").

As a result, new approaches such as Zero Trust Network Architecture (ZTNA) have emerged (see the sidebar, "ZTNA"). ZTNA calls for distrusting all traffic unless a security policy explicitly says otherwise.

Micro-segmentation is one of the core concepts within the ZTNA model. It involves identifying traffic flows between workloads in an application and allowing only those necessary for the application to function.

WARNING

Imposing ZTNA boils down to segmenting the network (although in a fine-grained manner) and controlling traffic flow between segments. Security teams should implement ZTNA by configuring their internal firewalls. Unfortunately, not all internal firewall designs are flexible enough to enable a ZTNA implementation.

FLAT NETWORKS

A *flat network* is one that is not adequately segmented. A genuinely flat network has no internal segmentation at all, but contains hundreds or thousands of workloads. There are no restrictions on communication between the workloads.

An organization may end up with a flat network due to ad-hoc growth or lack of adequate network and security planning. After all, a flat network is simple to conceive and understand. New workloads can be added relatively easily, creating an incentive to put off the segmentation design work.

However, a flat network is also difficult to troubleshoot and secure. When something goes wrong on the network, any of the workloads might be a suspect. Such a network is notoriously difficult to defend because the compromise of a single workload exposes all the rest to attack.

ZTNA

A *Zero Trust Network Architecture* (ZTNA) avoids automatic access to any network resource without first verifying that the entity requesting access to it has the appropriate permissions.

With ZTNA, access to a network resource is not automatically granted based solely on the requesting entity's location. Thus, a workload cannot communicate with another workload just because both workloads are in the same sub-network.

Further, ZTNA forces the use of least-privileged access. For example, users in the human resources (HR) group are granted access to an HR application without simultaneously allowing access to users outside the HR group.

Finally, ZTNA requires the inspection of all traffic. Thus, to implement ZTNA in a data center, an organization must inspect all east-west traffic, not just the traffic that comes into the data center from an entity outside the data center.

Security teams achieve a ZTNA by micro-segmenting their network. When networks are segmented at the level of workloads and security policies applied to each micro-segment, access to a network resource must be explicitly permitted by the micro-segmentation policy. All traffic is automatically inspected to enforce the policy.

IN THIS CHAPTER

- Understanding how enterprise edge firewalls came to be used as internal firewalls
- Appreciating the lack-of-fit between enterprise edge firewall designs and east-west traffic

Chapter 3
Looking at the Status Quo for Internal Firewalls

Most internal firewalls in use today have practically the same software and hardware design as enterprise edge firewalls. Such an evolution would be fine if enterprise firewall designs were a good fit for east-west traffic. As this chapter shows, the fit leaves a lot to be desired.

Using Enterprise Edge Firewalls as Internal Firewalls

REMEMBER

There was a time when most organizations only had an enterprise edge firewall. The notion of internal firewalls did not exist yet.

As information technology (IT) became more mainstream, organizations expanded their networks. Most physical servers running heavy-duty applications were segregated from end-user devices such as desktops, laptops, and printers, and moved to data centers.

Most IT organizations maintained tighter control and scrutiny over their data centers than over end-user devices. End-user devices and end-users themselves were seen as easier targets than the servers. Cybercriminals often targeted end-user devices for infiltration first, and then used their foothold on the end-user device to access servers in the data center.

Thus, IT organizations were forced to segment their end-user networks from their server networks. The only firewalls available to carry out such a segmentation were the high-end enterprise edge firewall appliances. These firewalls were repurposed to function as internal firewalls, which is the status quo in most organizations today.

The enterprise edge firewall vendors realized that a new firewall market segment had been created. However, since customer organizations were already buying enterprise edge firewalls for use as internal firewalls, the firewalls vendors did not make any radical changes to their firewall designs.

Uncovering Problems with Enterprise Edge Firewalls

Application architectures have also evolved from the time when data center networks were segmented from end-user networks. Applications used to be monolithic but have since been split into tiers, with each tier running as one or more workloads. More recently, applications have been further modularized — they are now composed of multiple workload types and microservices in the organization's data center or the public cloud.

REMEMBER

Enterprise edge firewalls were never designed to protect tiered applications, let alone modern applications using microservices and containers. As a result, enterprise edge firewalls have fallen behind in their ability to protect applications in the data center.

Viewing a lack of granular enforcement

For example, in a three-tier application an internal firewall permits traffic between the web tier and the business-logic (app) tier of the application and between the app tier and database tier of

the same application. However, it blocks the traffic from the web tier to the database tier because this traffic should not exist in the ordinary course of operations.

That is, the granularity of enforcement required from an internal firewall is much higher than that needed for an edge firewall. In the preceding example, a typical enterprise edge firewall doesn't know that the three tiers belong to the same application, or that within that application, some traffic is permitted while other traffic is not.


WARNING

It's acceptable for an enterprise edge firewall to block traffic based on ports, protocols, and network (IP) addresses or to identify and block traffic to or from a specific consumer application such as Skype. However, an internal firewall must operate at a granular level — that of individual workloads within an application.

Looking at an inspection capacity mismatch

If an organization uses an enterprise edge firewall for east-west traffic and wants to inspect all or most of the traffic, it must deploy many firewalls to meet its throughput requirements. The number of firewalls can significantly increase the cost and complexity of the network security infrastructure.

As east-west traffic grows, so does the capacity requirement for the enterprise edge firewall. Every so often, the security team has to upgrade the firewall appliances — an expensive and disruptive proposition.


WARNING

Centralized inspection of north-south traffic using an enterprise edge firewall doesn't typically create performance bottlenecks because the volume isn't nearly as large as it is for east-west traffic. However, most organizations have significantly more east-west traffic than north-south and require higher capacity.

In practice, most organizations using enterprise edge firewalls to monitor east-west traffic don't inspect most of it because the cost and constraints of doing so are too high. East-west traffic does not cross the centralized firewall and remains uninspected, as shown in Figure 3-1.

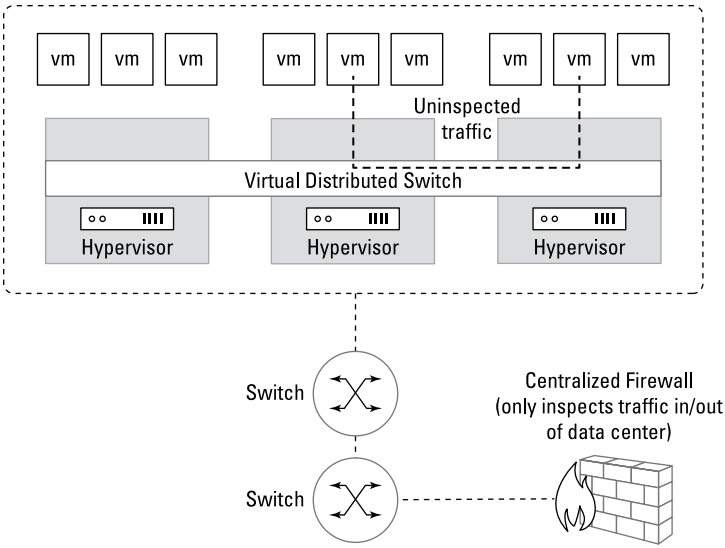

FIGURE 3-1: Uninspected east-west traffic.

Dealing with traffic hairpinning

When an enterprise edge firewall is used to monitor east-west traffic, traffic is forced to and back from a centralized appliance. This approach creates a hairpin pattern, which uses additional network resources (as shown in Figure 3-2) and increases the latency for demanding multi-tier applications.

In addition to increasing latency, hairpinning internal network traffic adds complexity, both from a design and an operations perspective.

WARNING

Networks must be designed to consider the additional (hairpinning) traffic routed through an enterprise edge firewall. From the operational side, the security team must adhere to the network design and be aware of constraints when sending additional traffic for inspection to the firewall.

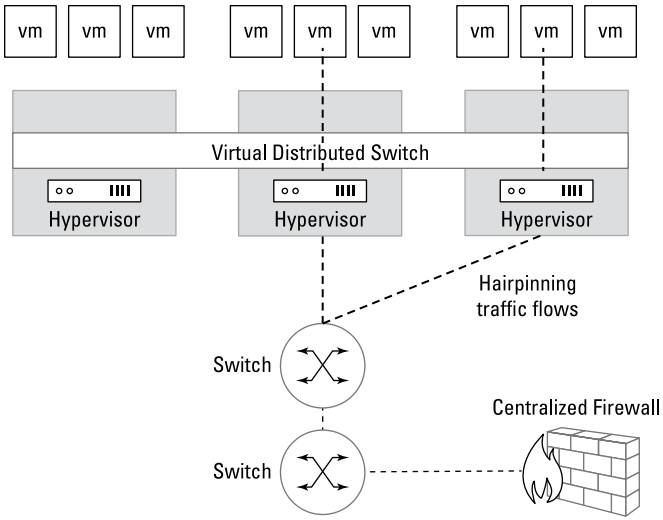

FIGURE 3-2: Traffic hairpinning.

Uncovering a lack of application topology visibility

WARNING

Monitoring east-west traffic and enforcing granular policies requires visibility down to the workload level. Standard enterprise edge firewalls deployed as internal firewalls do not have clear visibility into the communication patterns between the workloads and microservices that make up modern applications.

This lack of visibility is often because the firewall is not in the path of the east-west traffic. Even when the firewall is in the traffic path, it lacks the logic to distinguish traffic flows of one application, made up of multiple workloads, from those of another. The lack of visibility into application flows makes it challenging to create and enforce rules at the workload or individual traffic flow level.

An internal firewall should automatically determine the communication pattern between workloads and microservices, make security policy recommendations based on the pattern, and check that the resulting traffic flows match the deployed policies.

Handling policy lifecycle and mobility management

Traditional firewall management planes are designed to handle dozens of discrete firewalls and are not intended to support workload mobility with the automatic reconfiguration of security policies.

When an enterprise edge firewall is used as an internal firewall, security teams must manually create new security policies whenever a new workload is created and modify them when a workload is moved or decommissioned. Such policy creation is inherently difficult because enterprise edge firewalls require policies constructed from ports, protocols, and IP addresses rather than intrinsic attributes of workloads such as virtual machine names.

Workload mobility, as shown in Figure 3-3, is common in modern data centers.

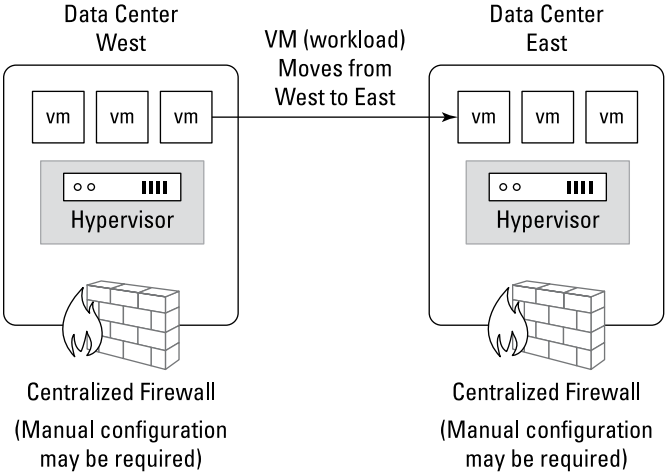

FIGURE 3-3: Workload mobility between data centers.

For example, an IT team may need additional capacity for some of the workloads. The capacity may only be available in a different data center than where the workloads are currently located.


TIP

A "VMotion" moves the workloads from one physical server to another but doesn't move the security policies associated with the workloads on the enterprise edge firewalls. Those must be adjusted manually.

Some applications and their constituent workloads may be decommissioned occasionally. Since the act of decommissioning applications is independent of the enterprise edge firewall, the security team must manually purge all security policies relevant to the decommissioned applications.

Security teams are often afraid of breaking existing traffic flows during a manual purge and choose not to remove outdated rules. As a result, old rules build up in the enterprise edge firewall, making future security management more difficult.

IN THIS CHAPTER

- Understanding why micro-segmentation solutions came about
- Appreciating the limitations of micro-segmentation orchestrators

Chapter 4
Evaluating Micro-Segmentation

Micro-segmentation solutions were developed in response to the mismatch between enterprise edge firewalls (used as internal firewalls) and the security team's desire to protect applications and their constituent workloads at a granular level. Micro-segmentation solutions are designed to enable distributed granular enforcement of security policies. As this chapter shows, most of these solutions come with limitations of their own.

Reviewing Micro-Segmentation Orchestrators

The typical micro-segmentation solution is made up of two main components, as shown in Figure 4-1.

The first is an agent that resides on the workload (for example, inside the virtual machine or the physical server hosting a particular database). The second is an orchestrator that computes security policies and distributes the policies to the agents for enforcement.

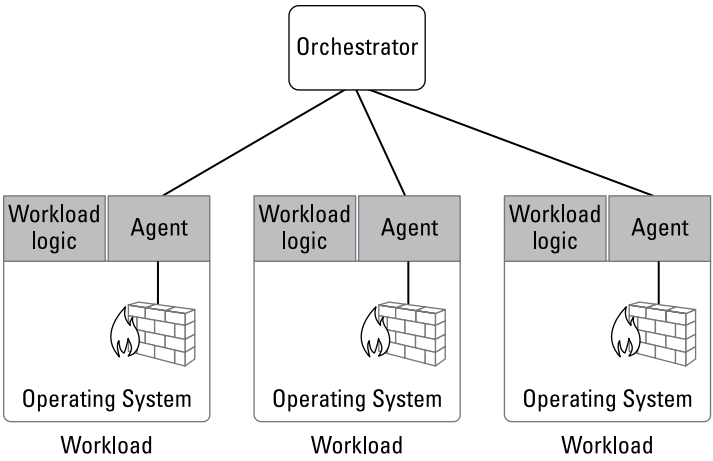

FIGURE 4-1: Micro-segmentation orchestrator and host firewalls.

REMEMBER

Most micro-segmentation solutions are designed to narrowly provide a mechanism for enforcement of security policies using network (IP) addresses, port numbers, and protocol identifiers in the header of an IP packet, and not much else.

TIP

This book refers to these narrow micro-segmentation solutions as *micro-segmentation orchestrators.* This term distinguishes the concept of micro-segmentation from the class of products that narrowly implement the concept.

Recognizing Problems with Orchestrators

The agents in micro-segmentation orchestrators are user-level processes that use the host operating systems' firewall to program security policies into the operating systems' kernel. The use of the operating system's firewall creates some significant problems, which the next section describes.

TIP

Micro-segmentation orchestrators are typically locked into the capabilities of the operating system's firewall. Because the operating systems used in a modern data center are maintained by third parties (for example, Red Hat and Microsoft), no straightforward path exists for micro-segmentation orchestrators to add security capabilities beyond those that the operating systems offer.

Network defense collapsed into endpoint

Security teams prefer to separate their network defense mechanisms from their endpoint defense mechanisms while constraining their security architecture's aggregate complexity. The separation of network defense from endpoint defense increases the resilience of the organization's security architecture — even if one of the defense systems is partially compromised, the team can use the other to contain attackers. At the same time, security teams prefer to deploy a small number of defense systems rather than multiple point defenses to keep their defense systems' total complexity manageable.

WARNING

If the micro-segmentation orchestrator is deployed instead of an internal firewall, the network defense system effectively collapses into the endpoint. This is because the micro-segmentation orchestrator uses the host operating system's stock firewall, and not an independent firewall, to secure network traffic. An attacker who manages to compromise an endpoint can easily disable both the operating system's firewall and the endpoint defense system. On the other hand, if a micro-segmentation orchestrator is used in addition to an internal firewall, the aggregate complexity of the organization's security architecture increases.

REMEMBER

Micro-segmentation orchestrators put security teams in a difficult spot. Either security teams must compromise on the resilience of their security architecture, or they must compromise on the simplicity of their security architecture.

No application-based or user-based policies

Often, security teams want to express their security policies at the level of applications and users. For example, a security team might want to implement a security policy whereby users in the human resources (HR) group are allowed access to HR applications but not finance applications.

WARNING

No simple way exists to create such a policy with micro-segmentation orchestrators. Most host operating system firewalls can only operate on IP address, port number, and protocol identifiers. Because micro-segmentation orchestrators are restricted to the host operating system's firewall capabilities, no

straightforward mechanism is available to express security policies for applications and users.

No threat controls

Security teams often need to deploy threat controls such as intrusion detection/prevention systems (IDS/IPS) to provide a second layer of protection within permitted traffic (see the sidebar, "IDS/IPS"). In some cases, government or industry regulations mandate the use of IDS/IPS, and in others, internal organizational policies require it.

WARNING

Because most host operating systems do not implement IDS/IPS, micro-segmentation orchestrators have no mechanism to provide an IDS/IPS as a second defensive layer. As a result, security teams are backed into purchasing and deploying a specialized IDS/IPS appliance or maintaining an internal firewall with IDS/IPS capability (in addition to the micro-segmentation orchestrator).

IDS/IPS

Intrusion detection/prevention systems (IDS/IPS) are software or network hardware deployed to analyze live traffic as it passes through the network. IDS/IPS deployments detect threats that have slipped through access control implemented by a firewall or micro-segmentation solution.

Regular-expression engines are the workhorses of IDS/IPS functionality. These engines are programmed to look for traffic patterns indicative of threats using a configuration language. Security teams refer to the patterns expressed using the IDS/IPS configuration language as *signatures.*

Further, most IDS/IPS implement protocol decoding engines to check conformance between the transiting traffic and published network protocol specifications.

Finally, some IDS/IPS detect abnormal traffic using statistical techniques. Such traffic can be indicative of ongoing attacks.

IN THIS CHAPTER

- Understanding the capabilities of distributed internal firewalls
- Recognizing a VMware NSX Service-defined Firewall

Chapter 5
Reimagining Internal Firewall Architecture

By themselves, neither enterprise edge firewalls nor micro-segmentation orchestrators are a good fit for internal firewalling. However, both types of solutions have some useful attributes that can be creatively combined, as this chapter shows. The result is a distributed internal firewall that matches the demands of east-west traffic.

Distributing Processing Engines

Internal firewalls need a distributed architecture to handle the scale of east-west traffic. They need distributed engines for all the packet processing that they do, including inspection for the following:

- Access control (traditional firewalling)
- Threat control (for intrusion detection/prevention, as an example)
- Analytics (for traffic analysis, as an example)

If the distributed processing engines are moved to the physical servers running the workloads, the processing is located close to the origin or destination of east-west traffic. This proximity of the processing engines to the workload has two significant effects:

- **No traffic hairpinning:** Because the processing engines are co-located with the workload, traffic between workloads does not traverse the network to and from a centralized appliance. As a result, the distributed processing engines eliminate traffic hairpinning, as shown in Figure 5-1.
- **Elastic inspection capacity matched to east-west traffic:** Because the distributed processing engines are housed on the same physical servers running the workloads, the processing capacity grows or shrinks with the servers.

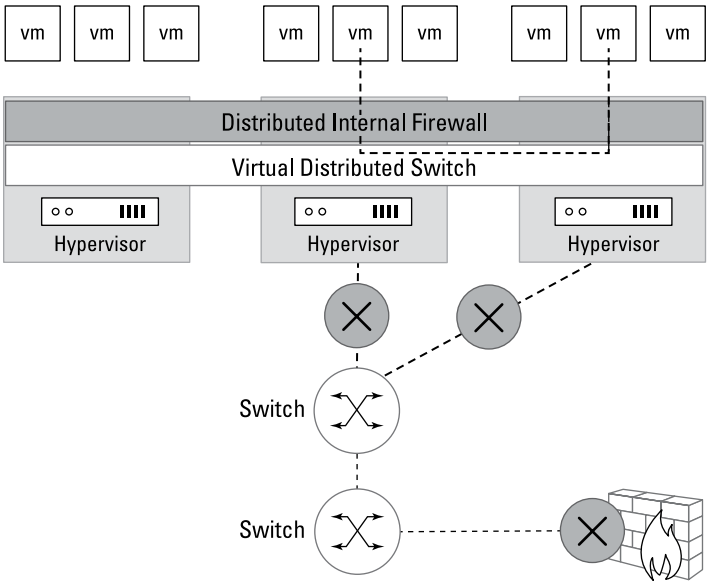

FIGURE 5-1: Avoiding traffic hairpinning with a distributed internal firewall.


TIP

More workloads typically mean more physical servers, which means more processing power for the distributed engines. Thus, security teams are freed from the constraints of the inspection capacity available in a centralized firewall appliance.

Distributing the processing engines comes with a responsibility to minimize the processing and memory-consumption overhead on the physical servers. A real-world implementation must ensure that it can saturate the physical connection (usually an Ethernet link) with a modest impact on the physical server.

Centralizing Management

REMEMBER

The usual difficulty with a distributed system is that of management. Managing a large number of distributed processing engines is a challenge. Fortunately, there is an architecturally sound solution to the problem of managing distributed engines: *centralized management.*

A centralized manager enables security teams to manage east-west network security from a single portal. While the engines are distributed, the central manager constructs a unified view of the engines for the security team's consumption. The centralized manager doesn't enforce any of the policies directly. Instead, it communicates the relevant policies to the distributed processing engines for enforcement.

If the centralized management and the distributed engines associate security policies with workloads, policy lifecycle management simplifies significantly. For example, policies can be created and designated for a workload even before the workload is created. When the workload is instantiated, it automatically inherits the appropriate security policies.

As a result, the workload is protected from the moment it comes into being. Should the workload move, the policy remains associated with the workload. If the workload decommissions, its associated policy goes dormant (see the sidebar, "Waxy Build-Up of Firewall Rules"), releasing computational power back to the server.

Finally, suppose the management system implements "security tags" associated with properties of entities on the network, such as operating system types, virtual machine names, user identities/groups, and application identities. In that case, security policies become simple to express and maintain. The security policies also become reconfigurable. For example, a development workload can be shifted to a product security zone by changing the security tag associated with that workload.

WAXY BUILD-UP OF FIREWALL RULES

Stale firewall rules build up for several reasons. First, the applications or the network may have changed, and the security team may not have had a chance to safely update the rules. Frequent application changes are common in data centers as workloads are updated and workload capacity is matched to performance requirements. Second, urgent IT requests result in temporary rules that the security team may not have the time to revisit. Third, the firewall policy model may be unintuitive, making firewall rule updates unnecessarily time-consuming, deterring clean-up.

Allowing the build-up of firewall rules affects the performance of the firewall because the firewall has to process unnecessary rules. The build-up also affects security because access to the network may be left open unintentionally. Finally, stale firewall rules make rule creation and modification more difficult because the security team is forced to understand unnecessary rules.

Accessing the Network Layer

REMEMBER

Even when an internal firewall has distributed processing engines and a central manager, it can't separate network defense from endpoint defense unless the firewall's distributed engines are independent of the endpoint.

When a distributed internal firewall has access to the network layer (the firewall can intercept and process packets from the traffic flow), it can accommodate the separation of network and endpoint defense.

The internal firewall must be in the traffic path to inspect a traffic flow. When the internal firewall has trusted access to the network layer, it can be inserted into the network infrastructure such that traffic passes through the firewall's processing engines. Thus, access to the network layer enables the internal firewall to create a network defense layer independent of endpoint defenses. Note that the operating system's firewall engine is not used by the internal firewall.

Making Traffic Information Portable

Suppose a workload is moved from one physical server to another. The underlying network will redirect the traffic to the new location of the workload. The new site will instantiate a new set of distributed processing engines to secure traffic flows.

However, the distributed processing engines will not work correctly on redirected traffic unless they have access to the traffic information (the traffic state) from just before the workload's move. If the traffic information is associated with the workload such that the traffic information moves with the workload, then the processing engines obtain the information they need to secure the redirected traffic.

REMEMBER

An internal firewall can't support secure workload mobility unless it has a mechanism to move information associated with existing traffic flows with the workload.

Secure workload mobility needs both the continuous delivery of traffic from the network and the continuous inspection of traffic from the distributed internal firewall. The continuous inspection itself becomes feasible when both of the following conditions exist:

- Centralized management and the distributed processing engines work together to maintain the association of security policies with workloads.
- Traffic information is made portable by associating it with workloads.

Evaluating VMware NSX Service-defined Firewall's Architecture

In the past, the cost, management complexity, and inspection capacity of enterprise edge firewalls deployed as internal firewalls have made for impossible economics. The expense of using enterprise edge firewalls to inspect 100 percent of the east-west traffic would destroy the budget.

Security policy management would also be a nightmare with a significant amount of manual intervention needed to create and maintain security policies using network (IP) addresses, port numbers, and protocol identifiers. Distributed internal firewalls change the economics of internal firewalling, making it possible for all east-west traffic to be inspected without breaking the budget or leading the security team into a security management nightmare.

VMware's NSX Service-defined Firewall is a faithful software-only implementation of a distributed internal firewall, as shown in Figure 5-2.

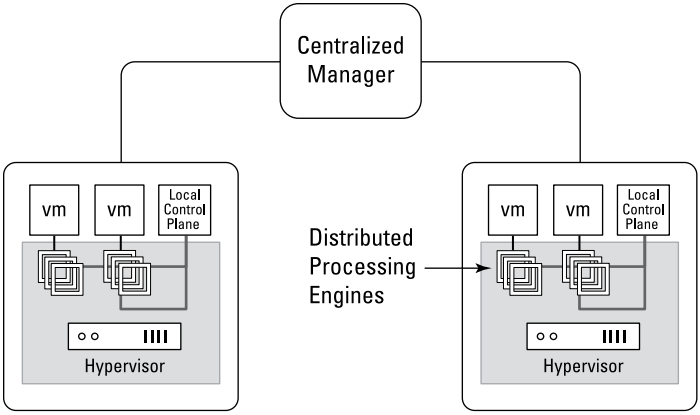

FIGURE 5-2: NSX Service-defined Firewall's distributed design.

It includes:

- **Distributed processing engines:** All the processing engines in the Service-defined Firewall — access control, threat control, and analytics — are distributed. The engines are implemented to extract the highest performance from physical servers (although the actual performance depends on the physical server).
- **Centralized management:** The Service-defined Firewall includes a centralized management system with an intuitive security policy model. Further, the central manager only communicates relevant policy changes to the distributed processing engines (via the local control plane). Finally, the firewall and its management system can be federated across multiple logical or physical deployments to support vast networks.

- **An intuitive policy model:** The central manager implements an abstract but intuitive security policy model. Policies can be described in terms of workload attributes such as operating system type, virtual machine names, user identities/groups, and application identities. Further, policies are associated with workloads, not the network, as workloads are the entities being protected. Abstract "security tags" can be associated with workloads to express otherwise complex policies simply. Finally, the policies are dynamic — they can be adjusted on-the-fly — enabling incremental changes to an organization's security posture.
- **Access to the network layer:** The Service-defined Firewall has access to the network layer, enabling it to separate network defense from endpoint defense (see sidebar, "NSX Service-Defined Firewall and Network Overlay").
- **Portable traffic information:** The Service-defined firewall associates traffic information with the workload. As a result, when a workload is moved, the traffic information moves with it, and ongoing traffic flows stay protected. No connection or traffic is dropped during the move. As importantly, no manual updates to security policies are needed to protect the moving workload.

NSX SERVICE-DEFINED FIREWALL AND NETWORK OVERLAY

A *network overlay* is a logical network, typically built atop a physical network. It uses the underlying physical network for transporting traffic but encapsulates packets from source workloads inside its own protocol header.

A network overlay provides additional services not available from the physical network. An example of such a service is the insertion of third-party networking or security products into the overlay network.

NSX and NSX Service-defined Firewall are implementations of two independent concepts and should not be confused. NSX implements a network overlay. NSX Service-defined Firewall implements a distributed internal firewall. The Service-defined Firewall works the same way, irrespective of whether a network overlay or a physical network is employed.

IN THIS CHAPTER

- Gaining insight into reincarnations of network security capabilities
- Recognizing how the architectural elements of VMware NSX Service-defined Firewall fit together

Chapter 6
Reimagining Internal Firewall Security Capabilities

An internal firewall with distributed processing engines, centralized management, access to the network layer, and portable traffic information is architecturally sound. However, even a sound architecture needs some smart logic inside its processing engines to fully enable security teams to protect against lateral movement. This chapter covers some of the additional logic required to convert a distributed firewall into a complete internal firewall.

Examining Distributed Access Control

Security teams want to deploy security policies for entities that they interact with daily. Applications and users (or user groups) are two such entities. It is a lot more intuitive to create access control policies — who can access what — by referencing applications and users rather than network (IP) addresses, port numbers,

and protocol identifiers, as is the case with micro-segmentation orchestrators. For example, a security team might want to create a policy that allows users in the finance group to access finance applications but not HR applications.

A proper implementation of a distributed internal firewall should automatically recognize the application from the traffic flow. Such an implementation also ought to identify users and the user groups they belong to, automatically.

A distributed firewall can implement its distributed access control engine to take advantage of its access to the network layer (the engine's position in the traffic path) and integrate with the server virtualization and active directory (AD) infrastructure in the data center. If it does so, the firewall can automatically discover the workload types at the two ends of a traffic flow — for example, an enterprise resource planning (ERP) software deployment communicating with a database. Similarly, the firewall can find AD user groups associated with users.


TIP

Thus, traffic flow information (state) maintained by the firewall includes user and application information, enabling the distributed internal firewall to inspect the traffic based on user groups and applications.

Examining Distributed Analytics

Most enterprise edge firewalls have some analytics capabilities. Although these capabilities remain available when enterprise edge firewalls are used as internal firewalls, most security teams deploy a parallel infrastructure for traffic analysis to augment their firewall. This parallel infrastructure is needed because the enterprise edge firewall does not see all the east-west traffic. Example infrastructure includes network traffic access, packet capture, and packet aggregation devices and out-of-band performance, and security monitoring tools.

On the other hand, a distributed internal firewall does see all the east-west traffic. Thus, a distributed internal firewall can absorb the analytics functionality within the firewall if it implements a

distributed analytics engine to accompany the distributed access control engine. The traffic analysis functionality in the analytics engine can be configured to detect symptoms of ongoing attacks, such as unusual network traffic spikes.

After an analytics facility is in place inside a distributed internal firewall, it can be used to create a map of traffic flows, enabling security teams to gain visibility into the behavior of applications and their constituent workloads. If the internal firewall also has a machine learning facility, it can automatically create security policy recommendations based on the observed traffic patterns. Finally, the internal firewall can run additional algorithms to highlight traffic that is not adequately secured, visually.

Examining Distributed Threat Control

Experienced security teams understand that threats often travel inside allowed traffic. The presence of threats inside permitted traffic is the motivation for the second defensive layer in addition to access control.

Intrusion detection and prevention systems (IDS/IPS) are a popular form of threat control used extensively in medium and large organizations. Traditional IDS/IPS, whether standalone or as part of a centralized enterprise edge firewall, are in the path of many traffic flows. Thus, they must turn on thousands of threat detection signatures to provide coverage across all the traffic flows. The number and type of signatures enabled affects IDS/IPS latency and throughput performance and false positive (spurious alert) rates. As a result, security teams spend considerable time tuning their IDS/IPS.

A distributed internal firewall can incorporate a distributed IDS/IPS (or more generally, threat control) engine. Because the engines are distributed, each engine need only run the signatures applicable to the workload the engine is protecting. Thus, only a small fraction of the signature set is turned on at a workload, leading to a reduction in false positives that security teams must handle.

Evaluating NSX Service-defined Firewall's Security Capabilities

TIP

VMware's NSX Service-defined Firewall is a purpose-built internal firewall implemented entirely in software. The Service-defined Firewall includes the following distributed capabilities:

- Access control engine (a classic stateful firewall that also recognizes applications and users/user groups)
- Analytics engine
- Threat control engine (IDS/IPS)

A few additional traits beyond the security capabilities are relevant when comparing the Service-defined Firewall to micro-segmentation orchestrators.

First, micro-segmentation orchestrators tend to be point solutions brought in to solve a specific problem in a particular way. Typically, such solutions are bolted onto the information technology (IT) infrastructure. Such an approach is okay for some situations but is not a great general solution. In contrast, the Service-defined Firewall is built into the data center IT infrastructure with deep integrations with organization-wide technologies such as network overlay, server virtualization, and identity management systems.

Second, mature organizations often need advanced threat-hunting capabilities such as Network Traffic Analysis/Network Detection and Response (NTA/NDR) and sandboxing accompanied by a threat intelligence feed (see the sidebars, "NTA/NDR," "Sandboxing," and "Threat Intelligence"). The VMware network security portfolio includes both NTA/NDR and sandboxing (from VMware's Lastline Inc. acquisition). As of this writing, NTA/NDR and sandboxing are being integrated into the Service-defined Firewall. Most micro-segmentation solutions do not have access to similar threat-hunting technologies.

NTA/NDR

Network Traffic Analysis/Network Detection and Response (NTA/NDR) solutions monitor east-west traffic and traffic flow records. Typically, NTA/NDR solutions work by modeling regular traffic and using statistical and machine learning techniques to flag traffic outside the norm as anomalies. Subsequently, such solutions attempt to determine if the anomalies are threats.

Some NTA/NDR solutions can be configured to automatically respond to threats. Response actions include configuring firewalls to drop suspicious traffic flows and raising alerts in security dashboards.

SANDBOXING

Sandboxing solutions emulate operating environments (end-user and server systems), enabling organizations to probe suspicious files and other objects without fear of contaminating the rest of their IT infrastructure.

Typically, suspicious objects are automatically submitted to a sandbox for execution. The sandbox records the object's run-time behavior and uses various techniques, including statistical and machine learning methods, to determine whether the object is benign or harmful (that is, if the object is malware).

Most sandboxes also implement additional malware detection techniques such as static analysis of executable files and similarity comparison with previously encountered objects.

Sandboxes can be configured to just detect, or both detect and block malware objects from traffic flows.

THREAT INTELLIGENCE

Threat control systems such as IDS/IPS, NTA/NDR, and sandboxes need periodic updates to maintain their efficacy.

In the case of IDS/IPS, the discovery of new vulnerabilities leads to the release of new signatures. These signature updates are packaged into a threat intelligence feed from a vendor's cloud service to the IDS/IPS systems deployed by a customer.

For NTA/NDR and sandboxes, threat intelligence consists of IP addresses and network domains of command-and-control servers (used to direct malware's actions), malware distribution points, and toxic websites. Threat intelligence also includes characteristics and behaviors of malware and malware objects, such as files.

A vendor's threat intelligence feed can include other items such as detailed information on known vulnerabilities in information technology (IT) systems and reports on the attacks seen across the vendor's customer base.

IN THIS CHAPTER

- Gaining insight into physical and containerized workloads
- Recognizing workloads in the public cloud

Chapter 7
Looking Beyond the Virtualized On-Premises Data Center

Although many workloads in data centers are hosted on virtual machines (VMs), not all workloads are. An internal firewall should protect such workloads as well. This chapter shows how a distributed internal firewall extends beyond workloads hosted on VMs in virtualized data centers.

Extending to Physical Servers

Many data centers have a small number of legacy or specialized workloads that need to run directly on an operating system that, in turn, sits on the physical hardware. This arrangement is called *physical workloads* (workloads running on a physical server without an intermediate virtualization layer).

A distributed internal firewall can support physical workloads using three different techniques. They are:

- **Via virtualized workloads:** In the cases where the physical workload only communicates with virtualized workloads (workloads hosted on virtual machines), the distributed internal firewall can do all its processing on the virtualized-workload end of the traffic flow. No distributed processing engines reside on the physical server.
- **A gateway:** If the distributed internal firewall supports a gateway (or edge firewall) mode, a gateway is deployed between the workloads on physical servers and the rest of the data center. All the traffic that leaves the physical servers passes through the gateway, where the firewall processing is done on behalf of the physical server, as shown in Figure 7-1. Note that the gateway is a specialized extension of the internal firewall.

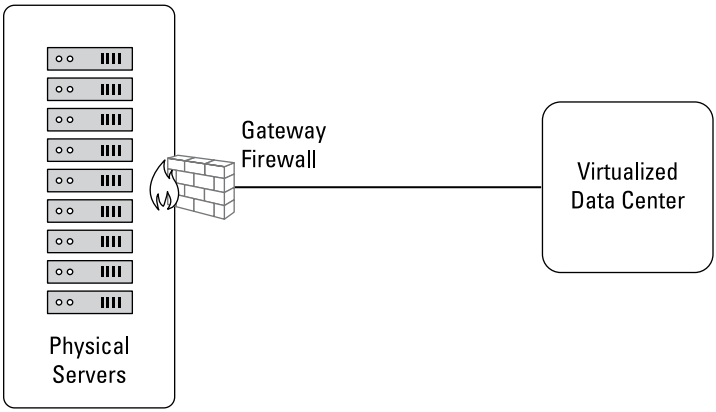

FIGURE 7-1: Gateway firewall for physical workloads.

- **A connector:** Finally, the distributed internal firewall can install a software connector (glue logic) directly on the physical server's operating system. In this case, the connector contains all the functionality of the distributed internal firewall. The firewall's view of the physical workload is almost the same as its view of a workload hosted on a virtual machine.

Extending to Containers

Containers are software packages that include the workload logic (say, a database) and everything else that the logic needs to run on an operating system. They have become popular as a method for creating workloads. For example, many organizations use Docker to containerize their workloads and Kubernetes to manage these workloads' deployment.

A distributed internal firewall can support containers in three ways:

- **Hosting inside a virtual machine:** In many cases, containers are hosted inside a virtual machine. In these cases, the distributed internal firewall considers containers inside the virtual machine as workloads. The distributed internal firewall needs to implement some additional logic to distinguish between the different containers inside a virtual machine. Other than that, not much else needs to change from the distributed internal firewall's point-of-view to support containers, as shown in Figure 7-2.

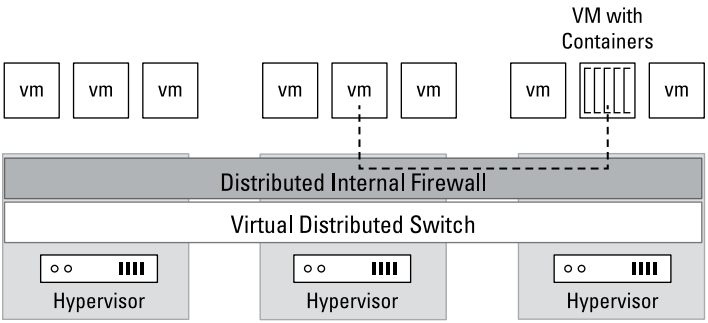

FIGURE 7-2: Containerized workloads inside virtual machines.

- **Replacing a virtual machine:** In some cases, containers run on the hypervisor (virtualization software) without an intermediary virtual machine. Here, containers replace virtual machines. The distributed internal firewall can support this arrangement if it is installed in the hypervisor. For the firewall, the containers appear as if they were virtual machines.

- **Residing on the physical server's operating system:** In other cases, containers run directly on the operating system that, in turn, runs on the physical server. That is, there is no intermediate hypervisor or virtual machine. The distributed internal firewall provides a connector for the operating system to make the containers on the operating system visible to the rest of the internal firewall deployment and carry out all the firewall processing.

Extending to Public Cloud

REMEMBER

Several public clouds have risen in prominence in recent years. These public clouds enable organizations to rent information technology infrastructure for their workloads. A distributed internal firewall can support public cloud workloads via two techniques:

- **Using public cloud controls:** If the security team wants to use the native firewall available from the public cloud provider, the distributed internal firewall provides a cloud gateway that translates security policy between the internal firewall and the public cloud firewall. In this case, the traffic inspection implementation in the public cloud differs from that in the data center, but the security policy unifies the differences, as shown in Figure 7-3.

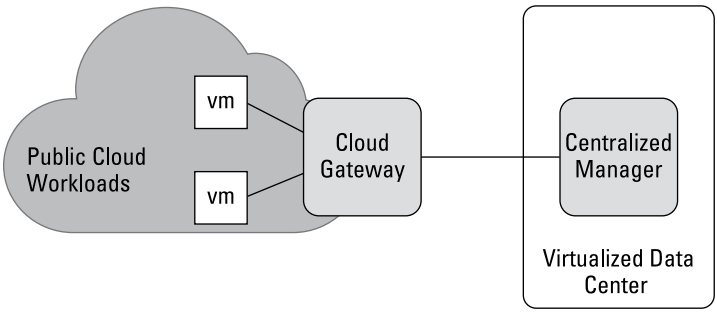

FIGURE 7-3: Internal firewall cloud gateway for public cloud workloads.

- **Using the distributed firewall's controls:** The distributed internal firewall can also deploy connectors with the public cloud workloads. These connectors contain all the processing capabilities of the distributed internal firewall, extending distributed internal firewalling to the workloads. In this case, the firewall controls provided by the public cloud are not used.

Supporting Physical Servers, Containers, and Public Cloud

As of this writing, VMware NSX Service-defined Firewall supports physical workloads, containerized workloads, and public cloud workloads. Further, all the modes that a distributed internal firewall could use for a particular type of workload are supported. They include:

- **Physical servers:** Protection is enabled via firewalling on the virtualized end of the traffic flow, a firewall gateway, or a software connector.
- **Containers:** Protection is made possible by hosting the container inside a virtual machine, directly running the container on the hypervisor, or by running the container on an operating system on a physical server. In all three cases, the Service-defined Firewall implements software connectors to enable traffic inspection.
- **Public cloud:** Protection is enabled by bringing the public cloud's security controls under the Service-defined Firewall's management or extending the Service-defined Firewall's distributed processing engines into the public cloud via software connectors.

Thus, the Service-defined Firewall enforces a uniform set of security policies across workloads, irrespective of the underlying infrastructure (on-premises or public cloud) or the workload type (physical server, virtual machine, or container).

IN THIS CHAPTER

- Macro-segmenting the network
- Protecting critical applications
- Gaining visibility to secure additional applications
- Securing all applications

Chapter 8
Ten (or So) Best Practices for Internal Firewalling

Implementing any new security approach requires time and commitment from a security team. For that reason, although protecting east-west network traffic is easier and faster with a distributed internal firewall, most organizations prefer to take a phased approach to improve data center security.

In addition to not overwhelming the security team with a significant initiative, breaking the deployment of an internal firewall into smaller projects delivers other benefits as well: It lets security teams prove success early and demonstrate the value of the approach to internal stakeholders. They can then choose to build on their experience to expand the use of distributed internal firewalling, gaining operational maturity, speed, and confidence as they progress.

The following steps have been used by VMware customers to start small and then continually strengthen their data center defenses.

Macro-Segment the Network

For many organizations, the first step in protecting east-west traffic is the most difficult. That's because attempting to *macro-segment* the network — segment it at a coarse level — using traditional, appliance-based firewalls has proven to be time-consuming, complex, and inflexible.

A frequent first step for the security team is to use the VMware NSX Service-defined Firewall to isolate and secure development, test, and production zones from each other, as shown in Figure 8-1. This segmentation immediately prevents attackers and malicious insiders from moving laterally between zones.

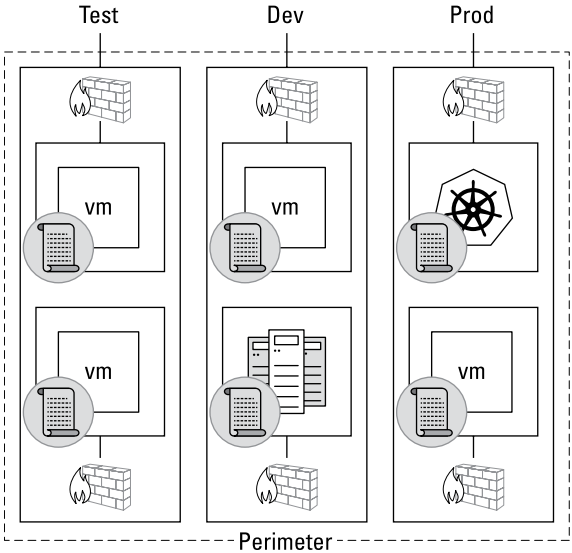

FIGURE 8-1: Macro-segmenting the network.

WARNING

When a security team tries to segment a "flat" network in the data center, it often must change the individual machines' network (IP) addresses. Further, the security team has to change their network architecture by using new or different virtual local area network identifiers (VLAN IDs) or physically changing the cabling on network switches and firewalls. Finally, the security team must accommodate the hairpinning traffic through the network.

TIP

Using a distributed internal firewall such as the Service-defined Firewall simplifies security architecture and accelerates time-to-value. No network address change or network re-design is needed, and traffic hairpinning is automatically prevented. Such an approach is also more flexible, adapting easily to changing network and security requirements as the business evolves.

Micro-Segment One Application

Typically, the next step to securing the data center is to start moving from macro-segmentation to micro-segmentation, enabling the security team to define and enforce more granular controls.

The security team chooses a well-understood and well-documented application critical to the business that should be isolated.

TIP

When considering a critical application to begin micro-segmentation, organizations often start with virtual desktop infrastructure (VDI).

VDI, while improving manageability, costs, and data protection for user desktops, exposes data center infrastructure to threats stemming from end-user security violations. However, using the Service-defined Firewall, the security team can isolate desktops from data center assets, protect the VDI infrastructure, and enable user-based access control, as shown in Figure 8-2. (See the sidebar, "Secure VDI with the NSX Service-defined Firewall").

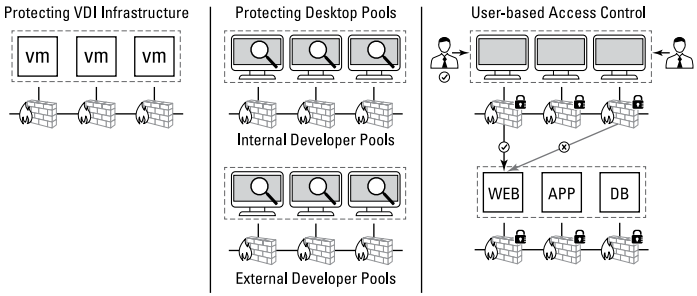

FIGURE 8-2: Protecting VDI environments.

Add IDS/IPS

The distributed intrusion detection/prevention (IDS/IPS) functionality of the Service-defined Firewall enables security teams to easily deploy threat controls for a layered security approach. No new hardware or software installation is required. Because the Service-defined Firewall is already deployed, the security team simply turns on IDS/IPS.

Protect Additional Well-Understood Applications

The security team builds on its experience, protecting the first application by choosing additional well-understood and well-documented applications critical to the business. Example applications include shared services such as Active Directory and Domain Name System (DNS).

Obtain East-West Traffic Visibility

As the security team gains more experience in operating a distributed internal firewall, it can expand its east-west traffic monitoring.

For applications that are not well-understood, the Service-defined Firewall gives the security team data center-wide visibility. It applies built-in machine learning to automate application discovery giving the security team a comprehensive map of application topography based on observed traffic flows, as shown in Figure 8-3.

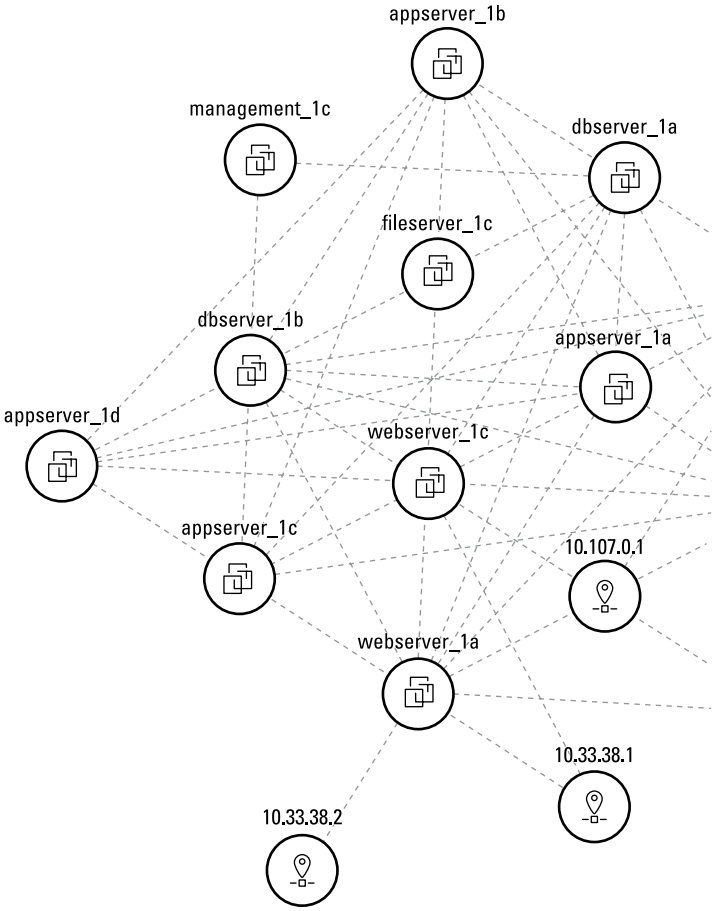

FIGURE 8-3: Visualizing east-west traffic flows.

Protect All Critical Applications

With visibility into east-west traffic, the security team can understand the behavior of the remaining critical applications. The security team can also use the Service-defined Firewall to automatically generate security policy recommendations, deploy updated policies, and monitor traffic flows for compliance against the policies.

Protect All Applications

The security team should now have the requisite experience and confidence to repeat the discovery-recommendation-deployment-compliance exercise for all the data center's remaining applications.

Widen IDS/IPS Deployment

As the security team uses the Service-defined Firewall across the data center, it will encounter pockets of servers and applications protected by standalone IDS/IPS appliances. Often the use of IDS/IPS is mandated by organizational or regulatory compliance requirements. Given the experience accumulated with IDS/IPS and data center-wide application protection in the preceding steps, the security team will find it straightforward to incrementally add IDS/IPS wherever needed. In the process, the security team can retire or repurpose the IDS/IPS hardware appliances.

Extend Beyond the Virtualized Data Center

Having secured the virtualized data center, it is time for the security team to also protect workloads hosted on physical servers and containers, and in the public cloud. The Service-defined Firewall treats all of these in much the same way as it does virtualized workloads.

Secure New Applications Before Deployment

The security team can get ahead of new application deployment by constructing and auditing policies even before the application is deployed. For example, the security team can create a new HRAPP tag for a new human resources (HR) application and create policies that block all communication unless they are between

entities with the HRAPP tag. Then, the still-dormant workloads in the application are associated with the HRAPP tag. Finally, the workloads are started up. The new application starts running and is secure from birth!

Proactively Hunt for Threats

Finally, the security team can get ahead of breaches by proactively looking for threats in their data centers. The security team can use Network Traffic Analysis/Network Detection and Response (NTA/NDR) solutions and sandboxes from the VMware network security portfolio toward this end. NTA/NDR and sandboxing technology came to VMware from its Lastline Inc. acquisition. As of this writing, both NTA/NDR and sandboxing are being integrated into the NSX Service-defined Firewall.

SECURE VDI WITH THE NSX SERVICE-DEFINED FIREWALL

Several attributes of the Service-defined Firewall contribute to securing VDI:

- **Distributed processing engines:** The Service-defined Firewall utilizes its access to the network layer and its distributed processing engines to inspect traffic right at the virtual desktops.
- **User-based policies:** Through its integration with Active Directory (AD), the Service-defined Firewall enables user-group specific security policies. User access to critical data center resources is governed by their AD group membership and access rights. The AD group information is associated with the virtual desktop workload, and user-based policies are enforced at the distributed processing engines of the virtual desktops.
- **Policy management model:** Security policies are based on an abstract policy model, using intuitive attributes such as operating system type, virtual machine names, and AD entries. This model eliminates dependencies on ephemeral network (IP) addresses and low-level traffic attributes while enabling virtual desktops' isolation with just a few policies.

(continued)

(continued)

- **Centralized management:** Security policies are defined centrally and distributed throughout the network. The central manager ensures policy consistency across virtual desktops and a hybrid network composed of virtual machines (VMs), containers, physical servers, and public cloud services.

Defend your network and infrastructure with distributed internal firewalls

Organizations can no longer rely on edge firewalls alone to provide network security. Once attackers get past the edge firewall, they can move laterally to high-value assets. This book discusses how internal firewalls can help your organization secure east-west network traffic and prevent attackers' lateral movement. It shows how distributed internal firewalls combine the best of hardware-based enterprise edge firewalls and software-based micro-segmentation solutions. It also describes the typical approach taken by organizations to successfully deploy distributed internal firewalls.

Inside...

- Learn why internal firewalls matter
- Recognize traffic and firewall types
- Understand micro-segmentation
- Reimagine firewall architecture
- Protect virtual machines, physical servers, and containers
- Adopt internal firewall best practices

Go to Dummies.com™
for videos, step-by-step photos, how-to articles, or to shop!

ISBN: 978-1-119-77296-5
Not For Resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.