Microsoft

Microsoft Azure Security Infrastructure

Yuri Diogenes
Dr. Thomas W. Shinder
Debra Littlejohn Shinder

Foreword by Mark Russinovich, Chief Technology Officer, Microsoft Azure

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Microsoft

Microsoft Azure Security Infrastructure

Yuri Diogenes
Dr. Thomas W. Shinder
Debra Littlejohn Shinder

PUBLISHED BY
Microsoft Press
A division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2016938684
ISBN: 978-1-5093-0357-1

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Support at mspinput@microsoft.com. Please tell us what you think of this book at http://aka.ms/tellpress.

Acquisitions and Developmental Editor: Karen Szall
Editorial Production: Online Training Solutions, Inc. (OTSI)
Technical Reviewer: Mike Toot; technical review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Jaime Odell (OTSI)
Indexer: Susie Carr (OTSI)
Cover: Twist Creative • Seattle

Contents

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can improve our books and learning resources for you. To participate in a brief survey, please visit:

http://aka.ms/tellpress

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can improve our books and learning resources for you. To participate in a brief survey, please visit:

http://aka.ms/tellpress

Foreword

Security is a critical requirement of any software system, but in today's world of diverse, skilled, and motivated attackers, it's more important than ever. In the past, security efforts focused on creating the strongest possible wall to keep attackers out. Security professionals considered the Internet hostile, and treated their own company or organization's systems as the trusted inner core, making relatively modest investments in segregating different environments and visibility into the interactions between different components. Now, the security world has adopted an "assume breach" mindset that treats perimeter networks as just one aspect of the protective pillar in a three-pillar approach that also includes detection and response. Attackers can and will penetrate the strongest defenses, and they can enter the network from inside. The perimeter is gone, and security architectures and investments are continuing to shift to address the new reality.

At the same time that the changing threat landscape is reshaping the approach to security, people have embarked on shifting their compute and data from infrastructure they deploy and maintain to that hosted by hyper-scale public cloud service providers. Infrastructure as a service (IaaS) and platform as a service (PaaS) dramatically increase agility by offering on-demand, elastic, and scalable compute and data. IT professionals and application developers can focus on their core mission: delivering compliant, standardized services to their organizations in the case of the former, and quickly delivering new features and functionality to the business and its customers in the latter.

You're reading this book because your organization is considering or has begun adopting public cloud services. You likely have already recognized that the introduction of the cloud provider into your network architecture creates new challenges. Whereas in your on-premises networks you use firewall appliances and physical routing rules to segregate environments and monitor traffic, the public cloud exposes virtualized networks, software load balancers, and application gateways, along with abstractions such as network security groups, that take their place. In some cases, the cloud offers services that give you insight and control that's either impossible or hard to achieve on-premises, making it easier to deliver high levels of security. The terminology, tools, and techniques are different, and creating secure and resilient "assume breach" cloud and hybrid systems requires a deep understanding of what's available and how to best apply it.

This book will serve as your trusted guide as you create and move applications and data to Microsoft Azure. The first step to implementing security in the cloud is knowing what the platform does for you and what your responsibility is, which is different depending on whether you're using IaaS, PaaS, or finished software services like Microsoft Office 365. After describing the differences, Yuri, Tom and Deb then move on to cover everything from identity and access control, to how to create a cloud network for your virtual machines, to how to more securely connect the cloud to your on-premises networks. You'll also learn how to manage keys and certificates, how to encrypt data at rest and in transit, how the Azure Security Center vulnerability and threat reporting can show you where you can improve security, and how Azure Security Center even walks you through doing so. Finally, the cloud and Internet of Things (IoT) are synergistic technologies, and if you're building an IoT solution on Azure, you'll benefit from the practical advice and tips on pitfalls to avoid.

The advent of the cloud requires new skills and knowledge, and those skills and knowledge will mean not only that you can more effectively help your organization use the cloud, but that you won't be left behind in this technology shift. With this book, you'll be confident that you have an end-to-end view of considerations, options, and even details of how to deploy and manage more secure applications on Azure.

— Mark Russinovich

CTO, Microsoft Azure
July 2016

Introduction

Regardless of your title, if you're responsible for designing, configuring, implementing, or managing secure solutions in Microsoft Azure, then this book is for you. If you're a member of a team responsible for architecting, designing, implementing, and managing secure solutions in Azure, this book will help you understand what your team needs to know. If you're responsible for managing a consulting firm that is implementing secure solutions in Azure, you should read this book. And if you just want to learn more about Azure security to improve your skill set or aid in a job search, this book will help you understand Azure security services and technologies and how to best use them to better secure an Azure environment.

This book includes conceptual information, design considerations, deployment scenarios, best practices, technology surveys, and how-to content, which will provide you with a wide view of what Azure has to offer in terms of security. In addition, numerous links to supplemental information are included to speed your learning process.

This book is a "must read" for anyone who is interested in Azure security. The authors assume that you have a working knowledge of cloud computing basics and core Azure concepts, but they do not expect you to be an Azure or PowerShell expert. They assume that you have enterprise IT experience and are comfortable in a datacenter. If you need more detailed information about how to implement the Azure security services and technologies discussed in this book, be sure to check out the references to excellent how-to articles on *Azure.com*.

Acknowledgments

The authors would like to thank Karen Szall and the entire Microsoft Press team for their support in this project, Mark Russinovich for writing the foreword of this book, and also other Microsoft colleagues that contributed by reviewing this book: Rakesh Narayan, Eric Jarvi, Meir Mendelovich, Daniel Alon, Sarah Fender, Ben Nick, Russ McRee, Jim Molini, Jon Ormond, Devendra Tiwari, Nasos Kladakis, and Arjmand Samuel.

Yuri: I would also like to thank my wife and daughters for their endless support and understanding, my great God for giving me strength and guiding my path, my friends and coauthors Tom and Deb Shinder, my manager Sonia Wadhwa for her support in my role, and last but not least, to my parents for working hard to give me education, which is the foundation that I use every day to keep moving forward in my career.

Tom and Deb Shinder: Writing—even with coauthors—is in some ways an isolated task. You sit down at the keyboard (or in today's high tech, alternative input environment, dictate into your phone or even scribble onto your tablet screen) alone, and let the words flow from your mind to the document. However, the formation of those words and sentences and paragraphs and the fine-tuning of them through the editing and proofing process are based on the input of many, many other people.

Because there are far too many colleagues, experts, and friends and family who had a role in making it possible for this book to come into being, we aren't going to even attempt to name them all here. You know who you are. From the family members who patiently waited while we finished up a chapter, delaying dinner, to the myriad of Azure professionals both within and outside of Microsoft, to the folks at Microsoft Press whose publishing expertise helped shape this collection of writing from three different authors with very different writing styles into a coherent whole, and most of all, to those who asked for and will read and (we hope) benefit from this book: we thank you.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

http://aka.ms/AzSecInfra/errata

If you discover an error that is not already listed, please submit it to us from the same page.

If you need additional support, email Microsoft Press Book Support at:

mspinput@microsoft.com

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to:

http://support.microsoft.com

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter at:

http://twitter.com/MicrosoftPress

This page intentionally left blank

CHAPTER 3

Azure network security

To understand Microsoft Azure network security, you have to know all the pieces and parts that are included. That means this chapter begins with a description and definition of all the features and services related to Azure networking that are relevant to security. For each feature, the chapter describes what it is and provides some examples to help you understand what the feature does and why it's good (or bad) at what it does. Some capabilities in Azure networking don't have a security story to tell, so the chapter leaves out those capabilities.

After the groundwork is laid and you have a better understanding of Azure networking, the chapter discusses Azure security best practices. These best practices are a compilation of things that you should do regarding Azure network security if they are appropriate to your deployment.

The chapter ends with a description of some useful patterns that you might want to use as reference implementation examples on which you can build your own solutions.

The goal of this chapter is to help you understand the "what's" and "why's," because if you don't understand those, you'll never get to the how's; if you implement the "how's" without understanding the "what's" and the "why's," you'll end up with the same "it sort of grew that way" network that you might have on-premises today. (If your network isn't like that, consider yourself exceptionally wise or lucky.)

To summarize, the chapter:

- Discusses the components of Azure networking from a security perspective.
- Goes over a collection of Azure networking best practices.
- Describes some Azure network security patterns that you might want to adopt for your own deployments.

One more thing before you venture into the inner workings of Azure networking: If you've been with Azure for a while, you're probably aware that Azure started with the Azure Service Management (ASM) model for managing resources. Even if you haven't been around Azure since the beginning, you're probably aware of the "old" and "new" portals (the "old" portal is now called the "classic" portal and the new portal is called the "Azure portal"). The classic portal uses the ASM model. The new portal uses the resource management model known as Azure Resource Management. This chapter focuses only on the Azure Resource Management model and the networking capabilities and behavior related to this model.

The reason for this is that the ASM model is being phased out and there is no future in it, so it would be best to migrate your ASM assets (if you have any) to the new Azure Resource Management model.

> ***MORE INFO*** **For more information about the differences between the ASM and Azure Resource Management models, read the article "Azure Resource Manager vs. classic deployment: Understand deployment models and the state of your resources" at *https://azure.microsoft.com/documentation/articles/resource-manager-deployment-model*.**

Anatomy of Azure networking

Azure networking has a lot of moving parts, and figuring out what these different parts do can be intimidating. The networking documentation on *Azure.com* focuses on the names of the products, and unfortunately these product names do not always make it easy for you to intuit the functionality of the product or feature. (Of course, this isn't just an Azure networking problem; you can go to any major cloud service provider's site and be assailed with the same problem.)

For this reason, this section is broken down into headings that focus on the capability you're interested in. For example, instead of providing the product name "Azure ExpressRoute" (which is explained later in detail), the heading for that networking capability is "Cross-premises connectivity." Because most people in networking know what that is, you don't need to try to figure it out from a product name. This format should help you understand what Azure has to offer in the networking arena.

This section describes the following Azure networking capabilities:

- Virtual network infrastructure
- Network access control
- Routing tables
- Remote access
- Cross-premises connectivity
- Network availability
- Network logging
- Public name resolution
- Network security appliances
- Reverse proxy

Virtual network infrastructure

Before getting into the Microsoft Azure Virtual Network itself, you should know that all servers that you deploy in Azure are actually virtual machines (VMs). This is important to understand, because some people new to the cloud might think that a public cloud service provider like Microsoft offers dedicated hardware servers as a service.

With the understanding that you use VMs to host servers in Azure, the question is, how do those VMs connect to a network? The answer is that VMs connect to an Azure Virtual Network.

Azure Virtual Networks are similar to virtual networks that have virtualization platform solutions, such as Microsoft Hyper-V or VMware. Hyper-V is used in Azure, so you can take advantage of the Hyper-V virtual switch for networking. You can think of the Hyper-V virtual switch as representing a virtual network interface that a VM's virtual network interface connects to.

One thing that might be different than what you use on-premises is how Microsoft isolates one customer's network from another customer's network. On-premises, you might use different virtual switches to separate different networks from each other, and that's perfectly reasonable. You can do that because you control the entire network stack and the IP addressing scheme on your network, in addition to the entire routing infrastructure. In Azure, Microsoft can't give each customer that level of control because Microsoft needs to reuse the same private IP address space among all the different customers, and Microsoft can't tell each customer which segment of the private IP address space to use for their VMs.

To get around this challenge, Microsoft takes advantage of the Windows Server software-defined networking stack—also known as "Hyper-V Network Virtualization" (HNV). With HNV, Microsoft can isolate each customer's network from other customer networks by encapsulating each customer's network communications within a generic routing encapsulation (GRE) head that contains a field that is specifically for the customer. This effectively isolates each customer's network from the others, even if different customers are using the same IP address schemes on their Azure Virtual Networks.

> ***MORE INFO*** **For more information about Hyper-V network virtualization, read the article "Hyper-V Network Virtualization Overview" on TechNet at *https://technet.microsoft.com/library/jj134230(v=ws.11).aspx*.**

Azure Virtual Network provides you with the following basic capabilities:

- IP address scheme
- Dynamic Host Configuration Protocol (DHCP) server
- Domain Name System (DNS) server

IP address scheme

Azure Virtual Networks require you to use private IP addresses (RFC 1918) for VMs. The address ranges are:

- Class A: 10.0.0.0/24
- Class B: 172.16.0.0/12
- Class C: 192.168.1.0/24

You should create an Azure Virtual Network before you create a VM, because all VMs need to be placed on an Azure Virtual Network. Just like with on-premises networking, you should carefully consider which IP address scheme you want to use, especially if you think you will connect your Azure Virtual Network to your on-premises network. In that scenario, you should make sure there is no overlap between the IP addresses you use on-premises and those you want to use on an Azure Virtual Network.

When you create an Azure Virtual Network, you'll typically choose a large block (or the entire Class A, B, or C range in the preceding list). Then you'll subnet that range, just as you do on-premises.

From a security perspective, you should think about how many subnets you need and how large to make them, because you'll want to create access controls between them. Some organizations use subnets to define security zones, and then create network access controls between the subnets by using Network Security Groups (which is explained later) or a virtual appliance.

Another type of addressing you should consider is public addresses. When you create a VM, a public address is assigned to that VM. Note that the public address isn't bound to the actual network interface (although it might appear that way when you see the description in the portal or read the documentation). The public IP address is the address that external users or devices can use to connect to the VM from over the Internet.

Similar to the IP addresses that are actually bound to the network interfaces on the VM itself (explained in the next section), you can assign either a dynamic or static public IP address to a VM.

Dynamic IP addresses on a public interface aren't as much of a problem as they might be on the internal network—that is to say, on the Azure Virtual Network itself. The reason for this is that DNS is used for Internet name resolution, and few (if any) users or devices are dependent on a static IP address to reach an Internet-reachable resource.

However, there might be situations where you need to use a static IP address on the Internet. For example, you might have network security devices that have access controls so that specific protocols or source IP addresses are allowed access only to specific IP addresses in Azure. When that is the case, you should take advantage of static public IP addresses.

Other scenarios where static public IP addresses might be used include the following:

- You've deployed applications that require communications to an IP address instead of a DNS name.
- You want to avoid having to remap DNS entries for publicly accessible resources on an Azure Virtual Network.

- Applications deployed on Azure or other public or private cloud networks need to use static addresses to communicate with your services on an Azure Virtual Network.
- You use SSL certificates that are dependent on a static IP address.

MORE INFO **To learn more about Azure Virtual Networks, read the article "Virtual Network Overview" at *https://azure.microsoft.com/documentation/articles/virtual-networks-overview*.**

DHCP servers

After you create an Azure Virtual Network and then place a VM on the network, the VM needs to have an IP address assigned to it to communicate with other VMs on the Azure Virtual Network (in addition to communicating to on-premises resources and even the Internet).

You can assign two types of IP addresses to VMs:

- Dynamic addresses
- Static addresses

Both types of addresses are managed by an Azure DHCP server.

Dynamic addresses are typically DHCP addresses that are assigned and managed by the Azure DHCP server. Like any other DHCP-assigned address, the VM's address is assigned from the pool of addresses defined by the address space you chose for your Azure Virtual Network.

In most cases, the address won't change over time and you can restart the VM and it will keep the same IP address. However, there might be times when the VM needs to be moved to another host in the Azure fabric, and this might lead to the IP address changing. If you have a server that requires a permanent IP address, then do not use dynamic addressing for that VM.

For VMs that perform roles requiring a static IP address, you can assign a static IP address to the VM. Keep in mind that you do not configure the NIC within the VM to use a static IP address. In fact, you should never touch the NIC configuration settings within a VM. All IP addressing information should be configured within the Azure portal or by using PowerShell Remoting in Azure.

Examples of VMs that might need dedicated addresses include:

- Domain controllers.
- Anything that needs a static address to support firewall rules you might configure on an Azure Virtual Network appliance.
- VMs that are referenced by hard-coded settings requiring IP addresses.
- DNS servers you deploy on an Azure Virtual Network (discussed in the next section).

Keep in mind that you cannot bring your own DHCP server. The VMs are automatically configured to use only the DHCP server provided by Azure.

MORE INFO **For more information on IP addressing in Azure, read the article "IP addresses in Azure" at *https://azure.microsoft.com/documentation/articles/virtual-network-ip-addresses-overview-arm*.**

DNS servers

You can use two primary methods for name resolution on an Azure Virtual Network:

- Azure DNS server
- Your own DNS server

When you create an Azure Virtual Network, you get a simple DNS server in the bargain, at no extra charge. This simple DNS server service provides you with basic name resolution for all VMs on the same Azure Virtual Network. Name resolution does not extend outside of the Azure Virtual Network.

The simple Azure Virtual Network DNS is not configurable. You can't create your own A records, SRV records, or any other kind of record. If you need more flexibility than simple name resolution, you should bring your own DNS server.

You can install your own DNS server on an Azure Virtual Network. The DNS server can be a Microsoft standalone DNS server, an Active Directory–integrated DNS server, or a non–Windows-based DNS server. Unlike the situation with DHCP servers on an Azure Virtual Network, you are encouraged to deploy your own DNS servers if you need them.

The bring-your-own-device (BYOD) DNS server is commonly used when you want to create a hybrid network, where you connect your on-premises network with your Azure Virtual Network. In this way, VMs are able to resolve names of devices on your on-premises network, and devices on your on-premises network are able to resolve names of resources you've placed on an Azure Virtual Network.

Network access control

Network access control is as important on Azure Virtual Networks as it is on-premises. The principle of least privilege applies on-premises and in the cloud. One way you do enforce network access controls in Azure is by taking advantage of Network Security Groups (NSGs).

The name might be a little confusing. When you hear "Network Security Group," you might think it's related to a collection of network devices that are grouped in a way that allows for common or centralized security management. Or maybe you'd think such a group might be a collection of VMs that belong to the same security zone. Both of these assumptions would be wrong.

A Network Security Group is the equivalent of a simple stateful packet filtering firewall or router. This is similar to the type of firewalling that was done in the 1990s. That is not said to be negative about NSGs, but to make it clear that some techniques of network access control have survived the test of time.

The "Group" part of the NSG name refers to a group of firewall rules that you configure for the NSG. This group of rules defines allow and deny decisions that the NSG uses to allow or deny traffic for a particular source or destination.

NSGs use a 5-tuple to evaluate traffic:

- Source and destination IP address
- Source and destination port
- Protocol: transmission control protocol (TCP) or user datagram protocol (UDP)

This means you can control access between a single VM and a group of VMs, or a single VM to another single VM, or between entire subnets. Again, keep in mind that this is simple stateful packet filtering, not full packet inspection. There is no protocol validation or network level intrusion detection system (IDS) or intrusion prevention system (IPS) capability in a Network Security Group.

An NSG comes with some built-in rules that you should be aware of. These are:

- **Allow all traffic within a specific virtual network** All VMs on the same Azure Virtual Network can communicate with each other.
- **Allow Azure load balancing inbound** This rule allows traffic from any source address to any destination address for the Azure load balancer.
- **Deny all inbound** This rule blocks all traffic sourcing from the Internet that you haven't explicitly allowed.
- **Allow all traffic outbound to the Internet** This rule allows VMs to initiate connections to the Internet. If you do not want these connections initiated, you need to create a rule to block those connections or enforce forced tunneling (which is explained later).

MORE INFO **To learn more about Network Security Groups, read the article "What is a Network Security Group (NSG)?" at *https://azure.microsoft.com/documentation/articles/virtual-networks-nsg*.**

Routing tables

In the early days of Azure, some might have been a bit confused by the rationale of allowing customers to subnet their Azure Virtual Networks. The question was "What's the point of subnetting, if there's no way to exercise access controls or control routing between the subnets?" At that time, it seemed that the Azure Virtual Network, no matter how large the address block you chose and how many subnets you defined, was just a large flat network that defied the rules of TCP/IP networking.

Of course, the reason for that was because no documentation existed regarding what is known as "default system routes." When you create an Azure Virtual Network and then define subnets within it, Azure automatically creates a collection of system routes that allows machines on the various subnets you've created to communicate with each other. You don't have to define the routes, and the appropriate gateway addresses are automatically assigned by the DHCP server–provided addresses.

Default system routes allow Azure VMs to communicate across a variety of scenarios, such as:

- Communicating between subnets.
- Communicating with devices on the Internet.
- Communicating with VMs that are located on a different Azure Virtual Network (when those Azure Virtual Networks are connected to each other over a site-to-site VPN running over the Azure fabric).
- Communicating with resources on your on-premises network, either over a site-to-site VPN or over a dedicated WAN link (these options are explained later in the chapter).

That said, sometimes you might not want to use all of the default routes. This might be the case in two scenarios:

- You have a virtual network security device on an Azure Virtual Network and you want to pump all traffic through that device. (Virtual network security devices are explained later in the chapter.)
- You want to make sure that VMs on your Azure Virtual Network cannot initiate outbound connections to the Internet.

In the first scenario, you might have a virtual network security device in place that all traffic must go through so that it can be inspected. This might be a virtual IDS/IPS, a virtual firewall, a web proxy, or a data leakage protection device. Regardless of the specific function, you need to make sure that all traffic goes through it.

In the second scenario, you should ensure that VMs cannot initiate connections to the Internet. This is different from allowing VMs to respond to inbound requests from the Internet. (Of course, you have to configure a Network Security Group to allow those connections.) Also ensure that all outbound connections to the Internet that are initiated by the VMs go back through your on-premises network and out your on-premises network security devices, such as firewalls or web proxies.

The solution for both of these problems is to take advantage of User Defined Routes. In Azure, you can use User Defined Routes to control the entries in the routing table and override the default settings.

For a virtual network security device, you configure the Azure routing table to forward all outbound and inbound connections through that device. When you want to prevent VMs from initiating outbound connections to the Internet, you configure forced tunneling.

MORE INFO **For more information about User Defined Routes, read the article "What are User Defined Routes and IP Forwarding?" at *https://azure.microsoft.com/documentation/articles/virtual-networks-udr-overview*. For more information about forced tunneling, read "Configure forced tunneling using the Azure Resource Manager deployment model" at *https://azure.microsoft.com/documentation/articles/vpn-gateway-forced-tunneling-rm*.**

Remote access (Azure gateway/point-to-site VPN/RDP/ Remote PowerShell/SSH)

One big difference between on-premises computing and public cloud computing is that in public cloud computing you don't have the same level of access to the VMs as you do on-premises. When you run your own virtualization infrastructure, you can directly access the VMs over the virtual machine bus (VMbus). Access through the VMbus takes advantage of hooks in the virtual platform to the VM so that you don't need to go over the virtual networking infrastructure.

This isn't to say that accessing a virtual machine over the VMbus is easy to achieve. There are strong access controls over VMbus access, just as you would have for any network-level access. The difference is that VMbus access for on-premises (and cloud) virtualization platforms is tightly controlled and limited to administrators of the platform. Owners of the virtual machines or the services that run on the virtual machines typically aren't allowed access over the VMbus—and if they are, this level of access is often temporary and can be revoked any time the virtualization administrators decide it's necessary.

When you have VMs on a cloud service provider's network, you're no longer the administrator of the virtualization platform. This means you no longer have direct virtual machine access over the virtualization platform's VMbus. The end result is that to reach the virtual machine for configuration and management, you need to do it over a network connection.

In addition to needing to go over a network connection, you should use a remote network connection. This might be over the Internet or over a dedicated WAN link. Cross-premises connectivity options (so-called "hybrid network connections") are explained in the next topic. This section focuses on remote access connections that you use over the Internet for the express purpose of managing VMs and the services running on the VMs.

Your options are:

- Remote Desktop Protocol (RDP)
- Secure Shell Protocol (SSH)
- Secure Socket Tunneling Protocol (SSTP)–based point-to-site VPN

Each of these methods of remote access depends on the Azure Virtual Network Gateway. This gateway can be considered the primary ingress point from the Internet into your Azure Virtual Network.

Remote Desktop Protocol

One of the easiest ways to gain remote access to a VM on an Azure Virtual Network is to use the Remote Desktop Protocol (RDP). RDP allows you to access the desktop interface of a VM on an Azure Virtual Network in the same way it does on any on-premises network. It is simple to create a Network Security Group rule that allows inbound access from the Internet to a VM by using RDP.

What's important to be aware of is that when you allow RDP to access a VM from over the Internet, you're allowing direct connections to an individual VM. No authentication gateways or proxies are in the path—you connect to a VM.

Like all simple things, using RDP might not be the best option for secure remote access to VMs. The reason for this is that RDP ports are often found to be under constant attack. Attackers typically try to use brute force to get credentials in an attempt to log onto VMs on Azure Virtual Networks. Although brute-force attacks can be slowed down and mitigated by complex user names and passwords, in many cases, VMs that are not compromised are considered temporary VMs and therefore do not have complex user names and passwords.

You might think that if these are temporary VMs, no loss or risk is involved with them being compromised. The problem with this is that sometimes customers put these temporary VMs on Azure Virtual Networks that have development VMs, or even production VMs, on them. Compromising these temporary VMs provides an attacker with an initial foothold into your deployment from which they can expand their breach. You don't want that to happen.

RDP is easy, and if you're sure that you're just testing the services and the VMs in the service, and you have no plans to do anything significant with them, then this scenario is reasonable. As you move from pure testing into something more serious, you should look at other ways to reach your VMs over the Internet. Other methods are described later in this chapter.

MORE INFO **For more information about more secure remote access that uses RDP and other protocols, read the article "Securing Remote Access to Azure Virtual Machines over the Internet" at *https://blogs.msdn.microsoft.com/azuresecurity/2015/09/08/securing-remote-access-to-azure-virtual-machines-over-the-internet*.**

Secure Shell Protocol

Remote Desktop Protocol and the Secure Shell Protocol (SSH) are similar in the following ways:

- Both can be used to access both Windows and Linux VMs that are placed on an Azure Virtual Network.
- Both provide for direct connectivity to individual VMs.
- User names and passwords can be accessed by brute force.

As with RDP, you should avoid brute-force attacks. Therefore, as a best practice, you should limit direct access to VMs by using SSH over the Internet. An explanation of how you can use SSH more securely is provided in the next section.

MORE INFO **For more information about how to use SSH for remote management of VMs located on an Azure Virtual Network, read the article "How to Use SSH with Linux and Mac on Azure" at *https://azure.microsoft.com/documentation/articles/virtual-machines-linux-ssh-from-linux*.**

SSTP-based point-to-site VPN

Although "point-to-site" VPN in relation to Azure might sound like a new VPN-type technology (sort of like how so-called "SSL-VPN" is not really a VPN in many cases), it's not new. Rather, it's a new name applied to traditional remote access VPN client/server connections, which has been around a long time. What makes point-to-site VPN special is the VPN protocol that's used, which is the Secure Socket Tunneling Protocol (SSTP).

The SSTP VPN protocol is interesting because, unlike other methods of remote access VPN client/server connections (such as IPsec, LTP/IPsec, or PPTP), the SSTP protocol tunnels communications over the Internet by using a TLS-encrypted HTTP header. What this means in practice is that SSTP can be used across firewalls and web proxies. Some people might find it funny to hear someone say that SSTP can be used to get across "restrictive firewalls" because it uses TCP 443 to connect to the VPN gateway server from your Azure Virtual Network. It sounds funny because, among network security and firewall experts, TCP port 443 is known as the "universal firewall port." That is to say, if you allow outbound TCP 443, you allow just about everything.

For those of you who are not networking experts, you should understand what a remote access VPN client/server connection is and how it works (from a high level).

When you establish a VPN connection, what you're doing is creating a virtual "link layer" connection. (Think of an Ethernet cable connection as a link-layer connection.) The amazing thing about VPN is that this link-layer connection actually happens over the Internet and you can use it to establish that connection with a VPN server. In the case of Azure, you're establishing that connection between your laptop and the Azure gateway.

The link-layer connection is like a virtual cable (referred to in this book as a "tunnel") and you can pass just about any kind of network traffic through that tunnel. This is useful because the tunnel is encrypted, so no one can see inside the tunnel because the traffic inside the tunnel moves over the Internet.

After the VPN connection is established between your laptop and the Azure VPN gateway, your laptop isn't connected to a specific Azure VM. Instead, your laptop is connected to an entire Azure Virtual Network, and with this connection, you can reach all the VMs on that Azure Virtual Network. This helps you make RDP and SSH connections more secure. But how does it do that?

The key here is that in order to establish the point-to-site VPN connection, you have to authenticate with the VPN gateway. The Azure VPN gateway and VPN client both use certificates to authenticate with each other. Certificate authentication isn't susceptible to brute-force attacks like direct RDP or SSH connections over the Internet can be. This is a nice security advantage.

The big advantage comes from the fact that you can run RDP or SSH traffic inside the SSTP VPN tunnel. After you establish the point-to-site VPN connection, you can start your RDP or SSH client application on your laptop and connect to the IP address of the VM on the Azure Virtual Network that you're connected to. Of course, you have to authenticate again to access the VM.

This means that you can block direct inbound access for the RDP and SSH protocols to VMs on your Azure Virtual Network over the Internet and still reach them by using those protocols after you establish the VPN connection. This entire process is secure because you have to authenticate the VPN connection first, and then authenticate again with the RDP or SSH protocols.

MORE INFO **To learn more about point-to-site connectivity between individual computers such as laptops and tablets, read the article "Configure a Point-to-Site VPN connection to a VNet using the classic portal" at *https://azure.microsoft.com/documentation/articles/vpn-gateway-point-to-site-create*.**

Cross-premises connectivity

The previous section explained how you can connect a single device like a laptop or tablet to an Azure Virtual Network to gain network access to all the VMs connected to that Azure Virtual Network. This section explains how you can connect an entire network to an Azure Virtual Network.

This introduces the topic of what is known as "cross-premises connectivity." Probably a better term would be "across sites" connectivity, but that doesn't sound as fancy. Regardless, what this term means is connectivity between two sites. The first site is usually your on-premises network (which is a network that your organization owns and controls) and an Azure Virtual Network. When cross-premises connectivity is enabled, you can pass traffic between the on-premises network and your Azure Virtual Network.

You can do this in two ways with Azure:

- Site-to-site VPN
- Dedicated WAN link

Site-to-site VPN

Site-to-site VPN is similar to the point-to-site VPN described earlier. Recall that with a point-to-site VPN, you can connect a single device (at a time) to an Azure Virtual Network. To be clear, that doesn't mean that when you use a point-to-site VPN you can only connect a single device at a time, which would block all other connections to the Azure Virtual Network. What it means is that when you use a point-to-site VPN, only that device is connected to the Azure Virtual Network. Other devices can connect to the same Azure Virtual Network by using a point-to-site VPN at the same time.

In contrast to a point-to-site VPN, with a site-to-site VPN, you can connect an entire network to an Azure Virtual Network. Site-to-site VPNs are sometimes called "gateway-to-gateway" VPNs because each end of the connection is a VPN gateway device.

VPN gateways are like routers. On a non-VPN network, a router is used to route packets to different subnets on your on-premises network. The routed connections go over Ethernet or wireless connections. A VPN gateway acts as a router too, but in the case of the VPN gateway, connections routed over the VPN gateway are not routed from one subnet to another subnet on your on-premises network. Instead, they are routed from your on-premises network to another network over the Internet by using a VPN tunnel. Of course, the remote network can also route packets back to your on-premises network.

When you use a site-to-site VPN with an Azure Virtual Network, you route packets to and from the Azure Virtual Network and your on-premises networks. You must have a VPN gateway on your on-premises network that works with the VPN gateway used by Azure. Most industry standard on-premises VPN gateways work with the Azure VPN gateway. Note that in contrast to the point-to-site VPN connections that use SSTP, the site-to-site VPN uses IPsec tunnel mode for the site-to-site VPN connection.

Using site-to-site VPN connections has a couple of downsides:

- Connections to Azure top out at around 200 megabits per second (Mbps).
- They, by definition, traverse the Internet, which could be a security issue.

The first issue really isn't a security problem, although it's related to performance and performance limitations, which can bleed into availability, which could lead to problems with the "A" in the confidentiality, integrity, and availability (CIA) triad of security. If you exceed your site-to-site VPN bandwidth and your users and devices can't get to what they need on your Azure Virtual Network, then you have a compromise in availability, and, hence, security issues. That is to say, you've essentially created a denial-of-service (DOS) attack on yourself because you chose a connectivity option that doesn't support your application and infrastructure requirements.

The second issue is more of a classic security problem. Any traffic that moves over the Internet will potentially be exposed to "hacking," "cracking," redirection, and other attempts to compromise the data. Although it is true that the site-to-site VPN uses a more secure IPsec tunnel that supports the latest cipher suites and modern encryption technologies, there is always the chance that if you have a dedicated attacker that wants your information, he will find weaknesses and compromise the data within the tunnel.

That said, if the attacker wants your data that much, he can find easier ways to get to it than to try and compromise your site-to-site VPN connection. But the possibility should be mentioned because the topic of this chapter and book is network security.

For environments that need the highest level of security and performance, you should review the option discussed in the next section, a dedicated WAN link.

MORE INFO **For more information about site-to-site VPN connectivity to Azure, read the article "Create a virtual network with a Site-to-Site VPN connection using the Azure classic portal" at *https://azure.microsoft.com/en-us/documentation/articles/vpn-gateway-site-to-site-create*.**

Dedicated WAN link

A dedicated WAN link is a permanent connection between your on-premises network and another network. With Azure Virtual Network, a dedicated WAN link provides a permanent connection between your on-premises network and an Azure Virtual Network. These dedicated WAN links are provided by telco providers and do not traverse the Internet. These connections are private, physical connections between your network and an Azure Virtual Network.

Microsoft provides you with the option to create a dedicated WAN link between your on-premises network and Azure Virtual Network by using ExpressRoute. (The name might change over time, so be sure to check the Azure Security Team blog on a regular basis.)

ExpressRoute provides you with:

- Up to 10 gigabits per second (Gbps) of connectivity between your on-premises network and an Azure Virtual Network.
- A dedicated, private connection that does not traverse the Internet.
- A service-level agreement (SLA) that guarantees uptime and performance.

As you can see, the level of performance you get with an ExpressRoute dedicated WAN link far exceeds what you get from a site-to-site VPN. That 10 Gbps is 50 times the maximum speed available with any site-to-site VPN you can establish to an Azure Virtual Network.

The security advantage is clear: the connection doesn't traverse the Internet and therefore isn't exposed to all the potential risks that are inherent in an Internet connection. Sure, someone might be able to gain access to the telco, but the odds of that happening are much lower than the security risks that you're exposed to on the Internet.

The SLAs are important. With a site-to-site VPN, you're depending on the Internet. The Internet doesn't have SLAs, and you get the best possible effort from all the telco providers and the network they manage. Your packets move over a number of networks and you hope for the best, but in no way can anyone guarantee you uptime or performance. That's how the Internet-at-large works.

With dedicated WAN links, the telco providers control the entire network. From your premises or co-location, the telco controls all traffic and performance across the channel. They can identify where problems are and fix them, and they can improve performance anywhere they want in the path. That's why dedicated WAN links are so efficient and expensive.

Note that ExpressRoute provides two different types of dedicated WAN links:

- Multiprotocol line switching (MPLS) to your on-premises network
- Exchange Provider connectivity, where the ExpressRoute connection terminates at a Telco Exchange Provider location

The MPLS version of ExpressRoute tops out at around 1 Gbps, whereas the Exchange Provider option provides you with up to 10 Gbps.

MORE INFO **To learn more about dedicated WAN links to Azure Virtual Networks, read the article "ExpressRoute Technical Overview" at *https://azure.microsoft.com/documentation/articles/expressroute-introduction*.**

Network availability

As explained earlier, the "A" in the CIA security triad is availability. From a network perspective, you should ensure that your services are always available and that you take advantage of network availability technologies. Azure has a few availability services that you can take advantage of:

- External load balancing
- Internal load balancing
- Global load balancing

External load balancing

To understand external load balancing, imagine that you have a three-tier application: a web front end, an application logic middle tier, and a database back end. The web front-end servers accept incoming connections from the Internet. Because the web front-end servers are stateless servers (that is to say, no information is stored on the servers that needs to persist beyond sessions), you deploy several of them. It doesn't matter which of these your users connect to, because they are all the same and they all forward connections to the middle-tier application logic servers.

To get the highest level of availability and performance from the web front-end server, you should ensure that all the incoming connections are equally distributed to each of the web front ends. You should avoid a situation where one server gets too much traffic. This kind of situation decreases application performance and possibly could make the application unavailable if that server becomes unavailable. To solve this problem, you can use external load balancing.

When you use external load balancing, incoming Internet connections are distributed among your VMs. For web front-end servers, external load balancing ensures that connections from your users are evenly distributed among those servers. This improves performance because no VM is handling an excessive load, and also improves uptime because if for some reason one or more VMs fail, other VMs you've configured the connections to be load balanced to are able to accept the connections.

MORE INFO **To learn more about external load balancing, read the article "Load balancing for Azure infrastructure services" at *https://azure.microsoft.com/documentation/articles/virtual-machines-linux-load-balance*.**

Internal load balancing

External load balancing is used for incoming connections from the Internet. If you refer back to the example three-tier application discussed earlier, you might want to load balance the other tiers in the solution. The application logic tier and the database tiers are different from the web front ends because they are not Internet facing. These tiers do not allow incoming connections to the Internet. In fact, it's likely that you'll configure them so that these other tiers are only allowed to communicate with VMs that they must communicate with.

For example, the application logic tier needs to accept incoming connections from the web front ends, and no other service (for the time being, ignore the discussion about management access). In this case, you configure a Network Security Group so that the application logic tier VMs can accept only incoming connections from the web front-end servers.

Similarly, the database tier VMs do not need to accept connections from the Internet or the web front ends; they need to accept incoming connections from the application logic VMs only. In this case, you configure a Network Security Group in such a way as to allow only incoming connections from an application logic tier.

That handles the connection security part of the puzzle, but you still have the availability component to deal with. The web front ends have external load balancing to help them out with that. But what about the application logic and database servers?

For these VMs, you should use internal load balancing.

Internal load balancing works the same way as external load balancing, with the difference being that the source and destination VMs are internal; no source or destination devices are on the Internet for internal load balancing. The source and destination can all be on Azure Virtual Networks, or on an Azure Virtual Network and an on-premises network.

> ***MORE INFO*** **To learn more about internal load balancing, read the article "Internal Load Balancer Overview" at *https://azure.microsoft.com/documentation/articles/load-balancer-internal-overview*.**

Global load balancing

With cloud computing in Microsoft Azure, you can massively scale your applications—so much so that you can make applications available to almost anybody in the world, and each user, regardless of location, can have great performance and availability.

So far, you've read about external and internal load balancing to improve availability. Although these technologies are critical to ensuring that your applications are always online, they suffer from the same limitation: they work on a per-datacenter basis. That is to say, you can only configure internal and external load balancing among VMs located in the same datacenter.

At first you might think that's not a big problem. The Azure datacenters are large, they have a huge reserve capacity, and if one or a thousand physical servers in the Azure datacenter go down, you'll still be up and running because of the built-in redundancy in the Azure fabric.

Although that's all true, you should consider what might happen if an entire datacenter becomes unavailable, or even an entire region. It's possible that the power might go out for a datacenter or an entire region due to some kind of natural or unnatural event. If your solutions are confirmed to a single datacenter or region, you might suffer from outages.

If you want to better secure yourself against these kinds of outages, you should design your applications to take advantage of the scalability of the cloud. Azure makes it possible for you to increase the scale by placing components of your applications all over the world.

The trick is to make sure that your users can access those applications. To do this, you should take advantage of something known as a "global load balancer." Global load balancers take advantage of the Domain Name System (DNS) to ensure that:

- Users access your service by connecting to the datacenter closest to that user. (For example, if a user is in Australia, that user connects to the Australian Azure datacenter and not one in North America.)
- Users access your service by connecting to the closest alternate datacenter if the closest datacenter is offline.
- Users have the best experience with your service by accessing the datacenter that is most responsive, regardless of the location of that datacenter.

Azure provides you with a global load balancer in the form of Azure Traffic Manager. With Azure Traffic Manager, you can:

- Improve the availability and responsiveness of your applications.
- Perform maintenance tasks or upgrade your applications and have them remain online and available by having users connect to an alternate location.
- Distribute and load balance traffic for complex applications that require specific load balancing requirements.

MORE INFO **To learn more about Traffic Manager and how you can take advantage of all its global load balancing features, read the article "What is Traffic Manager?" at *https://azure.microsoft.com/documentation/articles/traffic-manager-overview*.**

Network logging

It's standard practice to access network information from the network itself. You can do this in many ways, but typically enterprise organizations include some kind of network intrusion detection system (NIDS) inline so that all network traffic can be monitored. It's a matter of opinion regarding how valuable such devices are, given the large number of never-seen alerts that are generated, and if seen, that are never addressed.

Regardless, there is some value in having visibility at the network level. For that reason, many customers are interested in how they can get the same or similar level of visibility into network traffic on their Azure Virtual Network.

At the time this chapter was written, you can't get the same level of visibility into network traffic that you can get on-premises. Many of the on-premises devices work at the Link layer (OSI layer 2), which is not available on Azure Virtual Networks. The reason for this is that Azure Virtual Networks make use of software-defined networking and network virtualization, so the lowest level of traffic analysis you can get is at the Network layer (OSI layer 3).

It is possible to get network layer network information if you want to push all traffic through a virtual network security device. That is pretty easy to do for traffic destined to and from a particular network subnet, and the way you do it on an Azure Virtual Network is the same as you would do it on-premises: you ensure that the virtual network security device is in the path to the destination subnet by configuring the routing tables on your Azure Virtual Network. You do this by configuring User Defined Routes, which were described earlier in this chapter.

Although this is easy for inter-subnet communications, it's not easy if you want to see what's happening between two VMs on the same subnet on your Azure Virtual Network. The reason for this is that you can't easily take advantage of an intermediary virtual network security device between subnets, because the two VMs that are communicating with each other are on the same subnet. That doesn't mean it can't be done. You can install a VM on each subnet that acts as a proxy (web proxy and perhaps a SOCKS proxy). Then all communications are sent to the proxy, and the proxy forwards the connections to the destination host on the same subnet. As you can imagine, this can end up being complex and unwieldy if you have even a few subnets.

At this time, you have the ability to get some network information for traffic that moves through Network Security Groups. In particular, you can:

- Use Azure audit logs to get information about connections made through a Network Security Group.
- View which Network Security Group rules are applied to VMs and instance roles based on the MAC address.
- View how many times each Network Security Group rule was applied to deny or allow traffic.

Although this is much less than you can do on-premises, the situation will most likely change soon. In fact, by the time you read this chapter, you might be able to obtain network information and bring your level of access much closer to what you have on-premises. Be sure to check the Azure Security Blog on a regular basis, where that information will be shared with you when it becomes available.

MORE INFO **To learn more about how you can obtain logging information from Network Security Groups, read the article "Log Analytics for Network Security Groups (NSGs)" at *https://azure.microsoft.com/documentation/articles/virtual-network-nsg-manage-log*.**

Public name resolution

Although the Azure DNS service is not strictly a security offering and it doesn't necessarily connect to any specific security scenario, you should be aware that Azure has a DNS server.

You can configure DNS zones in Azure DNS. However, Azure does not provide DNS registrar services, so you'll need to register your DNS domain name with a commercial domain registrar.

> ***MORE INFO*** **To learn more about the Azure DNS service, read the article "Azure DNS Overview" at *https://azure.microsoft.com/documentation/articles/dns-overview*.**

Network security appliances

You've read about the option to use virtual network security appliances in a number of places in this chapter. A virtual network security appliance is a VM that you can obtain from the Azure Marketplace and is usually provided by an Azure partner. These VMs are similar or the same as the network security device VMs you might be using on-premises today. Most of the major network security appliance vendors have their offerings in the Azure Marketplace today, and new ones are added daily. If you don't find what you want today, be sure to check tomorrow.

> ***MORE INFO*** **To learn more about what virtual network security devices are available in the Azure Marketplace, on the Azure Marketplace home page (*https://azure.microsoft.com/en-us/marketplace*), enter security in the search box. You can find Azure security partners at *https://azure.microsoft.com/marketplace/?term=security*.**

Reverse proxy

The final Azure Network component to cover before moving on to the Azure network security best practices section is that of a reverse proxy. If you are relatively new to networking, or haven't delved into networking beyond what you needed to know, you might not be familiar with the concept of proxy or reverse proxy.

A proxy is a network device that accepts connections for other devices and then recreates that connection to forward the connection request and subsequent packets to the destination. The proxy device, as the name implies, acts on behalf of the computer that is sending the request or the response. The proxy sits in the middle of the communications channel and, because of that, can do many security-related "things" that can help secure your network and the devices within it.

The most popular type of proxy is the "web proxy." A web proxy accepts connections from a web proxy client (typically a browser configured to use the IP address of the web proxy as its web proxy). When the web proxy receives the request, it can inspect the nature of the request and then recreate the request on behalf of the web proxy client. When the destination website responds, the web proxy receives the response on behalf of the web proxy client, and it can inspect the response. After receiving the response, the web proxy client forwards the response traffic back to the client that made the original request.

Why are proxy devices useful in a security context? Some of the things they can do include the following:

- Require the requestor to authenticate before the proxy accepts and forwards the connection request.
- Inspect the destination URL to determine whether the destination is safe or dangerous; if it's dangerous, the proxy can block the connection.
- Look at request and response traffic to determine whether there is dangerous payload, such as viruses or other malware, and block the malware from being delivered.
- "Crack open" encrypted communications between client and destination server (such as SSL connections) so that malware, leaked data, and other information that shouldn't be crossing the proxy boundary is stopped at the proxy. This type of "SSL bridging" can significantly improve security, because attackers often hide what they're trying to accomplish by encrypting communications, which normally works because most communications are not subject to SSL bridging.

Proxy devices can do these things and a lot more. One could write a book about just proxies. But this book and chapter aren't about proxies, so don't dig deeper into them than necessary.

The reason to bring up proxies in this chapter is that Azure has a reverse proxy service that you can use to proxy connections to your on-premises resources. The reverse proxy service is called Azure Active Directory Application Proxy. You won't find this service in the list of Azure Active Directory products on *Azure.com*, and you won't find it in the table of contents. However, you'll learn about it in this book.

Before going any further, it's important that you understand the difference between a "forward proxy" and a "reverse proxy." A forward proxy accepts connections from clients on your on-premises network and forwards those connections to servers on the Internet (or on networks other than the one on which the clients are making the requests). In contrast, a reverse proxy is one that accepts connections from external clients and forwards them to servers on your on-premises network. For those of you with a lot of experience in this area, you recognize that this is a bit of an oversimplification, but it does describe in general the differences between a forward and reverse proxy.

Traditional reverse proxy devices are typically placed near the edge of your on-premises network. Servers such as mail servers and collaboration servers (like Microsoft Exchange and SharePoint) can be reached by Internet-based clients through the reverse proxy server. Microsoft used to have its own reverse proxy servers named Internet Security and Acceleration Server

(ISA Server) and Threat Management Gateway (TMG). Both those products were excellent but unfortunately were discontinued.

With that said, maintaining on-premises proxy servers can be a lot of work. If you don't manage them well, they can take down your services, which makes no one happy. What if you could hand the management, troubleshooting, and updating of your reverse proxy server to someone else and avoid all that hassle?

That's the core value of the public cloud, and the core value of using the Azure Application Proxy server instead of using an on-premises reverse proxy.

The Azure Application Proxy is already built into Azure, and you configure it so that when client systems want to request resources on your on-premises servers, they actually make the request to the reverse proxy on Azure. The Azure Application Proxy forwards those requests back to your on-premises servers.

Like most reverse proxy solutions, they add a measure of security. Here are some things you can do with the Azure Application reverse proxy service:

- Enable single sign-on for on-premises applications.
- Enforce conditional access, which helps you to define whether or not a user can access the application based on the user's current location (on or off the corporate network).
- Authenticate users before their connections are forwarded to the Azure Application Proxy.

MORE INFO **To learn more about the Azure Application Proxy, read the article "Publish applications using Azure AD Application Proxy" at *https://azure.microsoft.com/documentation/articles/active-directory-application-proxy-publish*.**

Azure Network Security best practices

At this point, you should have a good understanding of what Azure has to offer in the network security space. This chapter provided information about all the major components of Azure networking that have some kind of tie to security, and went over a number of examples so that you have context for each of the components. If you remember and understand everything you've read so far, consider yourself in the top 10 percent of the class when it comes to Azure networking.

Although understanding the various aspects of Azure networking is required to ensure that your deployments are secure, knowing what those aspects are and how they work is the first step. What you should do now is put that knowledge into action by learning a few best practices.

About best practices

The best practices I describe in this section are based on my 20-year experience with network security in general and my 5-year experience with Azure networking in particular. Of course, best practices are based on two things: the positive experience others have had using a specific practice and the confirmation that the best practices work across a number of environments. Understanding this is key, because it's important for you to understand that I didn't come up with these best practices on my own—I've learned from our customers, from the Microsoft field, and from the engineers who created the Azure networking technologies. Thus, these best practices represent an amalgam of multiple groups of people who are smarter than me—and now I'm sharing with you the results of my experiences.

Tom Shinder
Program Manager, Azure Security Engineering

One more thing about best practices: one size does not fit all. Although these best practices are good things to do in most cases, they aren't good things to do in all cases. You always have to consider the environment in which you're considering these best practices. Sometimes you won't need to use one of these best practices because they just don't apply. Use your best judgment and do what is best for your network.

This section covers the following Azure networking best practices:

- Subnet your networks based on security zones.
- Use Network Security Groups carefully.
- Use site-to-site VPN to connect Azure Virtual Networks.
- Configure host-based firewalls on infrastructure as a service (IaaS) virtual machines.
- Configure User Defined Routes to control traffic.
- Require forced tunneling.
- Deploy virtual network security appliances.
- Create perimeter networks for Internet-facing devices.
- Use ExpressRoute.
- Optimize uptime and performance.
- Disable management protocols to virtual machines.
- Enable Azure Security Center.
- Extend your datacenter into Azure.

Subnet your networks based on security zones

As mentioned earlier, in the section about Azure Virtual Networks, when you create a new Azure Virtual Network, you're asked to select an IP address space in the Class A, B, or C range. These Azure Virtual Network IP address ranges are large, so you should always create multiple subnets. This is no different than what you do on-premises today.

One thing you should think about is what IP address space you want to use on your Azure Virtual Network. If you plan to connect your on-premises network to one or more Azure Virtual Networks, you need to ensure that there are no IP address conflicts. That is to say, you have to ensure that the IP address ranges you select and the subnets you create on your Azure Virtual Networks do not overlap with what you have on-premises. If there is overlap, that would cause routing table conflicts, and traffic will not be routed correctly to your subnets on your Azure Virtual Networks.

After you decide on your IP address range for your Azure Virtual Network, the next step is deciding how you want to define your subnets. One approach is to define your subnets based on the roles of the VMs you intend to place on those subnets.

For example, suppose you have the following classes of services you want to deploy on an Azure Virtual Network:

- **Active Directory Domain Controllers** You want these to support domain-joined VMs on your Azure Virtual Network.
- **Web front-end servers** You use these to support your three-tier applications.
- **Application logic servers** You use these to support middleware functions for your three-tier applications.
- **Database servers** You use these as the database back ends for your three-tier applications.
- **Update servers** You use these servers to centralize operating system and application updates for the VMs on your Azure Virtual Network.
- **DNS servers** You use these to support Active Directory and non–Active Directory name resolution for servers on your Azure Virtual Network.

You could create just one big subnet and put all your VMs on the same subnet. However, that's not a great way to help you enable secure network access control. A better solution is to define subnets for each of these roles and then put each VM that supports these roles into a subnet created for each role. That leads you to putting the domain controllers on the domain controllers' subnet, the database servers on the database server subnet, the web front ends on the web front-ends subnet, and so on.

Not only does this help you keep track of where the various servers that participate in each role are located, it also makes it much easier to manage network access controls. For example, if you choose to use NSGs for network access control, you can create a set of rules that is appropriate for all the VMs on the particular subnet. If you need to put another VM on one of the subnets, you don't need to update the NSG rules, because the existing rules will support all the machines on the subnet because they perform the same roles.

To make this clearer, consider the following simple situation with two subnets:

- Web front-end subnet
- Application logic subnet

Only web front-end VMs go into the web front-end subnet, and only application logic VMs go into the application logic subnet.

The rules for the web front-end subnet might look like this:

- Allow inbound TCP port 443 from the Internet to all IP addresses on the web front-end subnet.
- Allow outbound TCP port 443 from the web front-end subnet to all IP addresses in the application logic server's subnet.

The rules for the application logic server subnet might look like this:

- Allow inbound TCP port 443 from the web front-end subnet to all IP addresses on the application logic server's subnet.
- Allow outbound TCP port 1433 from the application logic server's subnet to all IP addresses on the database server subnet.

With these basic rules in place, you can easily put more front-end web servers onto the front-end web server's subnet without having to make any changes in the Network Security Group rules. The same goes with the application logic server's subnet.

Use Network Security Groups carefully

Although Network Security Groups are useful for basic network access control, keep in mind that they do not provide you any level of application layer inspection. All you have control over is the source and destination IP address, source and destination TCP or UDP port number, and the direction to allow access.

Another thing to be aware of is that if you want to create restrictive access rules with Network Security Groups, you have to be aware of what you might inadvertently block. Here are a few examples:

- VMs need to be able to communicate with IP addresses specific to the host operating system to get DHCP information. If you block access to this host port (which you need to discover by checking an ipconfig on your VMs to see what IP address is being used by the DHCP server), then your VMs will not be able to communicate with the DHCP server and will not be assigned IP addressing information.

- The DHCP server not only assigns an IP address to the VMs; it also assigns a DNS server and a default gateway. The DNS server will be a host server IP address (that is to say, an IP address owned by the host server, not by you), and the default gateway will be an address on your Azure Virtual Network subnet. If you block access to these IP addresses, you won't be able to perform name resolution or reach remote subnets. Neither of these conditions leads to trouble-free performance.
- Another scenario you might not think of is communications outside of your Azure Virtual Network, but still within the Azure fabric itself; for example, when you encrypt Azure Virtual Machines by using Azure Disk Encryption. To encrypt your operating system and data disks, the VM needs to be able to reach the Azure Key Vault Service and an Azure Application (these are prerequisites for Azure Disk Encryption). If you lock down your NSGs too tight, you won't be able to reach the Key Vault or the Azure Active Directory application, and your VMs won't start because the disks can't be unencrypted.

These are just a few examples. The message is to test your NSG rules thoroughly before going into production. By thoroughly testing, you won't have to deal with nasty surprises that might turn a successful deployment into a painful experience.

Use site-to-site VPN to connect Azure Virtual Networks

Eventually, you might decide you want to move the majority of your on-premises services into the Azure public cloud. You are likely going to find that as your presence in Azure grows, so will your need to use multiple Azure Virtual Networks.

You might want more than one Azure Virtual Network for many reasons. Some examples include:

- You have multiple on-premises datacenters and you want to connect to Azure Virtual Networks that are closest to the datacenter.
- You want resources in one region to be able to communicate with resources in another region over the fastest route possible.

> ***NOTE*** **Communications over the Azure fabric are faster than looping back through your on-premises network or looping through the Internet.**

- You want to use different Azure Virtual Networks to manage different classes of services, or assign them to different departments, or even different divisions or subsidiaries within your company.

These are just three examples, but you can probably come up with more. The point is that Azure Virtual Networks can grow as quickly as your on-premises network has over time. And at some point, you're going to want to connect some of those Azure Virtual Networks to one another.

The best way to do this is to connect them to each other over a site-to-site VPN connection over the Azure fabric. The site-to-site connection between the Azure Virtual Networks is similar to the site-to-site connection you establish between your on-premises network and an Azure Virtual Network. The difference is that the entire communications path between the Azure Virtual Networks is contained within the highly optimized Azure fabric itself.

An alternative to this approach is to have the Azure Virtual Networks communicate with each other over the Internet. This approach has security and performance implications that make it inferior to site-to-site VPN over the Azure fabric. Another alternative is to loop back through your on-premises network and out through another gateway on your network. In most cases, this is also a less efficient and potentially less secure solution.

MORE INFO **For more information about how to create a site-to-site VPN connection between two Azure Virtual Networks, read the article "Configure a VNet-to-VNet Connection by using Azure Resource Manager and PowerShell" at *https://azure.microsoft.com/documentation/articles/vpn-gateway-vnet-vnet-rm-ps*.**

Configure host-based firewalls on IaaS virtual machines

This is a best practice on-premises and in the cloud. Regardless of what operating system you deploy in Azure Virtual Machines, you want to make sure that a host-based firewall is enabled, just as you do on-premises.

Another feature of the host-based firewall on Windows virtual machines is IPsec. Although IPsec for intranet communications isn't widely used, there's always a good reason to turn on IPsec—that reason being that no network can be trusted and, therefore, regardless of where that network is and who owns and operates it, you should always consider any network (wired and wireless) untrusted and untrustable.

The dichotomy of the "trusted corporate network" versus "untrusted non-corporate networks" sounded good in the past before the widespread use of the Internet. But with the collision of multiple trends, such as cloud computing, using your own device, multi-homed devices (wireless devices that connect to a corporate network and other wireless networks at the same time), and numerous portable storage devices of all shapes, sizes, and capacities, it's not realistic to think that there is a material difference between the innate security of your on-premises network and any other network, including the Internet.

Well, there might be, but to think and act otherwise puts you at more risk than you need to be. That's why you should use IPsec for all communications that aren't encrypted by some other method (such as HTTPS or encrypted SMB 3.0). You can use IPsec on Azure Virtual Network to authenticate and encrypt all wire communications between VMs on the Azure Virtual Network, in addition to communications between those VMs in Azure and any devices you have on-premises.

If you do choose to use IPsec, be careful not to block host ports responsible for DHCP and DNS resolution, in addition to the default gateway and any storage addresses your VM might need access to.

MORE INFO **To learn more about how to use IPsec for server and domain isolation, read the article "Server and Domain Isolation Using IPsec and Group Policy" at *https://technet.microsoft.com/library/cc163159.aspx*.**

Configure User Defined Routes to control traffic

When you put a VM on an Azure Virtual Network, you might notice that the VM can connect to any other VM on the same Azure Virtual Network, even if the other VMs are on different subnets. This is possible because there is a collection of system routes that are enabled by default that allow this type of communication. These default routes allow VMs on the same Azure Virtual Network to initiate connections with each other, and with the Internet (for outbound communications to the Internet only).

Although the default system routes are useful for many deployment scenarios, sometimes you might want to customize the routing configuration for your deployments. These customizations allow you to configure the next hop address to reach specific destinations.

You should configure User Defined Routes when you deploy a virtual network security appliance, which is described in a later best practice.

There are other scenarios where you might want to configure custom routes. For example, you might have multiple network security appliances that you want to forward traffic to on the same or other Azure Virtual Networks. You might even have multiple gateways you want to use, such as a scenario where you have a cross-premises connection between your Azure Virtual Network and your on-premises location, in addition to a site-to-site VPN that connects your Azure Virtual Network to another Azure Virtual Network or even multiple Azure Virtual Networks.

Just as in the on-premises world, you might end up requiring a complex routing infrastructure to support your network security requirements. For this reason, paying close attention to your User Defined Routes will significantly improve your overall network security. Of course, ensure that you document all your customizations and include the rationale behind each one you make.

MORE INFO **To learn more about User Defined Routes, read the article "What are User Defined Routes and IP Forwarding" at *https://azure.microsoft.com/documentation/articles/virtual-networks-udr-overview*.**

Require forced tunneling

If you haven't spent a lot of time in the networking space, the term "forced tunneling" might sound a little odd. In the context of Azure networking, it can sound odd even to those who have experience in networking.

To understand why the term might sound odd, it helps to understand where the term comes from. First, what is "tunneling"? As explained earlier in this chapter when we covered VPN technologies, tunneling is a way to move data through an encrypted channel. (For you network purists out there, yes, you can tunnel within non-encrypted protocols, but let's keep it simple here.) When you establish a VPN connection, you create an encrypted tunnel between two network devices. After the tunnel is established, information can travel more securely within that tunnel.

Now consider a common scenario that many have experienced. You're at a hotel room and need to create a VPN connection between your laptop and the VPN server at your company. You use whatever software you need to use to establish the VPN connection. After you establish the connection, you can access servers and services on the corporate network as though you are directly connected to the corporate network.

Let's say that you want to go to a non-corporate website. You open your browser, enter the address, and go to the site. Does that connection go over the VPN connection and out your corporate firewalls, and then back through your corporate firewalls and back to your laptop over the VPN connection for the response?

It depends.

If your computer is configured to allow split-tunneling when using the VPN connection, it means that your computer will access the site by going over the Internet—it will not try to reach the site by going through your VPN connection to the corporate network and out to the Internet through your corporate network firewalls.

Most organizations consider this a security risk because when split-tunneling is enabled, your computer can essentially act as a bridge between the Internet and the corporate network, because it can access both the Internet and your corporate network at the same time. Attackers can take advantage of this "dual connection" to reach your corporate network through your split-tunneling computer.

The term "split-tunneling" itself is a bit of a misnomer, because there's only a single "tunnel" here: the encrypted VPN tunnel to your corporate network. The connection to the Internet itself is not "tunneled." So technically, you don't have a "split tunnel"; you have a "dual connection." Regardless, sometimes names for things aren't rational, so you'll have to accept the industry standard name for this phenomenon.

Let's say that you don't want to deal with the risk of split tunneling when your users are connected to your corporate network over VPN. What do you do? You configure something called "forced tunneling." When forced tunneling is configured, all traffic is forced to go over the VPN tunnel. If you want to go to a non-corporate website, then that request is going to go over the VPN connection and over your corporate network to your corporate firewalls, and then the corporate firewalls will receive the responses and forward the responses back to you over the VPN

connection. There will be no "direct" connections to any Internet servers (with "direct" meaning that the connections avoid going over the VPN connection).

What does this have to do with Azure network security?

The default routes for an Azure Virtual Network allow VMs to initiate traffic to the Internet. This process can pose a security risk because it represents a form of split tunneling, and these outbound connections could increase the attack surface of a VM and be used by attackers. For this reason, you should enable forced tunneling on your VMs when you have cross-premises connectivity between your Azure Virtual Network and your on-premises network.

If you do not have a cross-premises connection, be sure to take advantage of Network Security Groups (discussed earlier) or Azure Virtual Network security appliances (discussed next) to prevent outbound connections to the Internet from your Azure Virtual Machines.

> ***MORE INFO*** **To learn more about forced tunneling and how to enable it, read the article "Configure forced tunneling using the Azure Resource Manager deployment model" at *https://azure.microsoft.com/en-us/documentation/articles/vpn-gateway-forced-tunneling-rm*.**

Deploy virtual network security appliances

Although Network Security Groups and User Defined Routes can provide a certain measure of network security at the network and transport layers of the OSI model, in some situations, you'll want or need to enable security at high levels of the stack. In such situations, you should deploy virtual network security appliances provided by Azure partners.

Azure network security appliances can deliver significantly enhanced levels of security over what is provided by network level controls. Some of the network security capabilities provided by virtual network security appliances include:

- Firewalling
- Intrusion detection and prevention
- Vulnerability management
- Application control
- Network-based anomaly detection
- Web filtering
- Antivirus protection
- Botnet protection

If you require a higher level of network security than you can obtain with network-level access controls, then you should investigate and deploy Azure Virtual Network security appliances.

> ***MORE INFO*** **To learn about what Azure Virtual Network security appliances are available, and about their capabilities, visit the Azure Marketplace at *https://azure.microsoft.com/marketplace* and search on "security" and "network security."**

Create perimeter networks for Internet-facing devices

A perimeter network (also known as a DMZ, demilitarized zone, or screened subnet) is a physical or logical network segment that is designed to provide an additional layer of security between your assets and the Internet. The intent of the perimeter network is to place specialized network access control devices on the edge of the perimeter network so that only the traffic you want is allowed past the network security device and into your Azure Virtual Network.

Perimeter networks are useful because you can focus your network access control management, monitoring, logging, and reporting on the devices at the edge of your Azure Virtual Network. Here you would typically enable distributed denial-of-service (DDoS) prevention, IDS and IPS, firewall rules and policies, web filtering, network antimalware, and more. The network security devices sit between the Internet and your Azure Virtual Network and have an interface on both networks.

Although this is the basic design of a perimeter network, many different perimeter network designs exist, such as back-to-back, tri-homed, and multi-homed.

For all high-security deployments, you should consider deploying a perimeter network to enhance the level of network security for your Azure resources.

MORE INFO **To learn more about perimeter networks and how to deploy them in Azure, read the article "Microsoft Cloud Services and Network Security" at *https://azure.microsoft.com/documentation/articles/best-practices-network-security*.**

Use ExpressRoute

Many organizations have chosen the hybrid IT route. In hybrid IT, some of the company's information assets are in Azure, while others remain on-premises. In many cases, some components of a service are running in Azure while other components remain on-premises.

In the hybrid IT scenario, there is usually some type of cross-premises connectivity. This cross-premises connectivity allows the company to connect their on-premises networks to Azure Virtual Networks. Two cross-premises connectivity solutions are available:

- Site-to-site VPN
- ExpressRoute

Site-to-site VPN represents a virtual private connection between your on-premises network and an Azure Virtual Network. This connection takes place over the Internet and allows you to "tunnel" information inside an encrypted link between your network and Azure. Site-to-site VPN is a secure, mature technology that is deployed by enterprises of all sizes. Tunnel encryption is performed by using IPsec tunnel mode.

Although site-to-site VPN is a trusted, reliable, and established technology, traffic within the tunnel does traverse the Internet. In addition, bandwidth is relatively constrained to a maximum of about 200 Mbps.

If you require an exceptional level of security or performance for your cross-premises connections, you should consider using Azure ExpressRoute for your cross-premises connectivity. ExpressRoute is a dedicated WAN link between your on-premises location or an Exchange hosting provider. Because this is a telco connection, your data doesn't travel over the Internet and therefore is not exposed to the potential risks inherent in Internet communications.

MORE INFO **To learn more about how Azure ExpressRoute works and how to deploy it, read the article "ExpressRoute Technical Overview" at *https://azure.microsoft.com/documentation/articles/best-practices-network-security*.**

Optimize uptime and performance

Confidentiality, integrity, and availability (CIA) make up the three factors for evaluating a customer's security implementation. Confidentiality is about encryption and privacy, integrity is about making sure that data is not changed by unauthorized personnel, and availability is about making sure that authorized individuals are able to access the information they are authorized to access. Failure in any one of these areas represents a potential security breach.

Availability can be thought of as being about uptime and performance. If a service is down, information can't be accessed. If performance is so poor as to make the data unavailable, then you can consider the data to be inaccessible. Therefore, from a security perspective, you should do whatever you can to ensure that your services have optimal uptime and performance. A popular and effective method used to enhance availability and performance is to use load balancing. Load balancing is a method of distributing network traffic across servers that are part of a service. For example, if you have front-end web servers as part of your service, you can use load balancing to distribute the traffic across your multiple front-end web servers.

This distribution of traffic increases availability because if one of the web servers becomes unavailable, the load balancer will stop sending traffic to that server and redirect traffic to the servers that are still online. Load balancing also helps performance, because the processor, network, and memory overhead for serving requests is distributed across all the load balanced servers.

You should consider employing load balancing whenever you can, and as appropriate for your services. The following sections discuss appropriateness situations. At the Azure Virtual Network level, Azure provides you with three primary load balancing options:

- HTTP-based load balancing
- External load balancing
- Internal load balancing

HTTP-based load balancing

HTTP-based load balancing bases decisions about which server to send connections to by using characteristics of the HTTP protocol. Azure has an HTTP load balancer named Application Gateway.

You should consider using Azure Application Gateway when you have:

- Applications that require requests from the same user or client session to reach the same back-end VM. Examples of this are shopping cart apps and web mail servers.
- Applications that want to free web server farms from SSL termination overhead by taking advantage of Application Gateway's SSL offload feature.
- Applications, such as a content delivery network, that require multiple HTTP requests on the same long-running TCP connection to be routed or load balanced to different back-end servers.

> ***MORE INFO*** **To learn more about how Azure Application Gateway works and how you can use it in your deployments, read the article "Application Gateway Overview" at *https://azure.microsoft.com/documentation/articles/application-gateway-introduction*.**

External load balancing

External load balancing takes place when incoming connections from the Internet are load balanced among your servers located in an Azure Virtual Network. The Azure External Load Balancer can provide you with this capability, and you should consider using it when you don't require the sticky sessions or SSL offload.

In contrast to HTTP-based load balancing, the External Load Balancer uses information at the network and transport layers of the OSI networking model to make decisions on what server to load balance connections to.

You should consider using External Load Balancing whenever you have stateless applications accepting incoming requests from the Internet.

> ***MORE INFO*** **To learn more about how the Azure External Load Balancer works and how you can deploy it, read the article "Get Started Creating an Internet Facing Load Balancer in Resource Manager using PowerShell" at *https://azure.microsoft.com/documentation/articles/load-balancer-get-started-internet-arm-ps*.**

Internal load balancing

Internal load balancing is similar to external load balancing and uses the same mechanism to load balance connections to the servers behind them. The only difference is that the load balancer in this case is accepting connections from VMs that are not on the Internet. In most cases, the connections that are accepted for load balancing are initiated by devices on an Azure Virtual Network.

You should consider using internal load balancing for scenarios that will benefit from this capability, such as when you need to load balance connections to SQL servers or internal web servers.

Global load balancing

Public cloud computing makes it possible to deploy globally distributed applications that have components located in datacenters all over the world. This is possible on Azure due to its global datacenter presence. In contrast to the load balancing technologies mentioned earlier, global load balancing makes it possible to make services available even when entire datacenters might become unavailable.

You can get this type of global load balancing in Azure by taking advantage of Azure Traffic Manager. Traffic Manager makes it possible to load balance connections to your services based on the location of the user.

For example, if the user is making a request to your service from the European Union, the connection is directed to your services located in a European Union datacenter. This part of Traffic Manager global load balancing helps to improve performance because connecting to the nearest datacenter is faster than connecting to datacenters that are far away.

On the availability side, global load balancing ensures that your service is available even if an entire datacenter becomes available.

For example, if an Azure datacenter becomes unavailable due to environmental reasons or outages such as regional network failures, connections to your service would be rerouted to the nearest online datacenter. This global load balancing is accomplished by taking advantage of DNS policies that you can create in Traffic Manager.

You should consider using Traffic Manager for any cloud solution you develop that has a widely distributed scope across multiple regions and requires the highest level of uptime possible.

Disable management protocols to virtual machines

It is possible to reach Azure Virtual Machines by using RDP and SSH protocols. These protocols make it possible to manage VMs from remote locations and are standard in datacenter computing.

The potential security problem with using these protocols over the Internet is that attackers can use various brute-force techniques to gain access to Azure Virtual Machines. After the attackers gain access, they can use your VM as a launch point for compromising other machines on your Azure Virtual Network or even attack networked devices outside of Azure.

Because of this, you should consider disabling direct RDP and SSH access to your Azure Virtual Machines from the Internet. With direct RDP and SSH access from the Internet disabled, you have other options you can use to access these VMs for remote management:

- Point-to-site VPN
- Site-to-site VPN
- ExpressRoute

Point-to-site VPN is another term for a remote access VPN client or server connection. A point-to-site VPN enables a single user to connect to an Azure Virtual Network over the Internet. After the point-to-site connection is established, the user is able to use RDP or SSH to connect to any VMs located on the Azure Virtual Network that the user connected to via point-to-site VPN. This assumes that the user is authorized to reach those VMs.

Point-to-site VPN is more secure than direct RDP or SSH connections because the user has to authenticate twice before connecting to a VM. First, the user needs to authenticate (and be authorized) to establish the point-to-site VPN connection; second, the user needs to authenticate (and be authorized) to establish the RDP or SSH session.

A site-to-site VPN connects an entire network to another network over the Internet. You can use a site-to-site VPN to connect your on-premises network to an Azure Virtual Network. If you deploy a site-to-site VPN, users on your on-premises network are able to connect to VMs on your Azure Virtual Network by using the RDP or SSH protocol over the site-to-site VPN connection, and it does not require you to allow direct RDP or SSH access over the Internet.

You can also use a dedicated WAN link to provide functionality similar to the site-to-site VPN. The main differences are:

- The dedicated WAN link doesn't traverse the Internet.
- Dedicated WAN links are typically more stable and performant.

Azure provides you with a dedicated WAN link solution in the form of ExpressRoute.

Enable Azure Security Center

Azure Security Center helps you prevent, detect, and respond to threats, and provides you with increased visibility into, and control over, the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Azure Security Center helps you optimize and monitor network security by:

- Providing network security recommendations.
- Monitoring the state of your network security configuration.
- Alerting you to network-based threats both at the endpoint and network levels.

It is highly recommended that you enable Azure Security Center for all of your Azure deployments.

> ***MORE INFO*** **Azure Security Center is covered in more detail in Chapter 7, "Azure resource management security."**

Extend your datacenter into Azure

Many enterprise IT organizations are looking to expand into the cloud instead of growing their on-premises datacenters. This expansion represents an extension of existing IT infrastructure into the public cloud. By taking advantage of cross-premises connectivity options, it's possible to treat your Azure Virtual Network as just another subnet on your on-premises network infrastructure.

However, many planning and design issues need to be addressed first. This is especially important in the area of network security. One of the best ways to understand how you approach such a design is to see an example.

Microsoft has created the Datacenter Extension Reference Architecture Diagram and supporting collateral to help you understand what such a datacenter extension would look like. This provides a reference implementation that you can use to plan and design a secure enterprise datacenter extension to the cloud. You should review this document to get an idea of the key components of a secure solution.

MORE INFO **For more information about the Datacenter Extension Reference Architecture Diagram, read "Datacenter extension reference architecture diagram – Interactive" at *https://gallery.technet.microsoft.com/Datacenter-extension-687b1d84*.**

This page intentionally left blank

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Z