Microsoft

Microsoft Azure Security Infrastructure

Yuri Diogenes
Dr. Thomas W. Shinder
Debra Littlejohn Shinder

Foreword by Mark Russinovich, Chief Technology Officer, Microsoft Azure

Microsoft

Microsoft Azure Security Infrastructure

Yuri Diogenes
Dr. Thomas W. Shinder
Debra Littlejohn Shinder

- 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*.**