Microsoft

Configuring Windows Server Hybrid Advanced Services

Exam Ref AZ-801

Orin Thomas

Microsoft

Exam Ref AZ-801 Configuring Windows Server Hybrid Advanced Services

Orin Thomas

will provide information on each server associated with it, including which updates are missing and the nature of those updates.

5. To deploy updates, select Schedule Update Deployment in the Update Management blade of the Azure console. When configuring a scheduled update, you need to provide the following information:
   - **Name of the update deployment** This is especially useful if you configure a recurring schedule.
   - **Operating system** Azure Update Management allows you to deploy updates to computers running Windows or Linux operating systems in a single update deployment, but not to both operating systems in one deployment.
   - **Groups** You can configure query-based groups so that the update deployment targets computers that meet the criteria that are specified in the query.
   - **Machines to update** Allows you to select specific computers to which the update deployment applies.
   - **Update classifications** Rather than select specific updates, you configure an update deployment so that all updates that meet a specific update classification will be deployed (though you do have the option of excluding specific updates by Update ID). For Windows computers, the update classifications are Critical Updates, Security Updates, Update Rollups, Feature Packs, Service Packs, Definition Updates, Tools, and Updates.
   - **Include/exclude updates** Allows you to choose specific updates based on KBIDs. You can find relevant KBIDS in the list of missing updates in the Azure Update Management console.
   - **Schedule settings** Allows you to specify when updates will be deployed. You can configure a schedule to recur. When you do this, all updates that meet the classification characteristics specified in the update deployment will be deployed.
   - **Maintenance window** The amount of time that can be taken to install updates, with the final 20 minutes of the assigned maintenance window reserved for restart operations. For example, if you set a maintenance window of 120 minutes, the update installation will be halted after 100 minutes have elapsed so that restart operations can occur.
   - **Reboot options** Allows you to specify whether the server can restart automatically after update deployment or whether this step must be performed manually.

Azure Update Management provides you with more telemetry about which servers need updates than WSUS does.

NEED MORE REVIEW? **UPDATE MANAGEMENT**

You can learn more about Azure Automation Update Management at *https://docs.microsoft.com/en-us/azure/automation/update-management/overview*.

Onboard Windows Server using Azure Automanage

Azure Automanage is a service that automates the deployment of a number of existing services used to secure and monitor Azure IaaS VMs, including those running the Windows Server operating system. Automanage will onboard and configure the following services for Windows Server IaaS VMs:

- Boot diagnostics
- Security Center
- Azure monitoring
- Azure Automation Update Management
- Azure backup
- Azure Log Analytics
- Azure configuration management
- Change tracking and inventory
- Azure Automation accounts

On-premises Azure Arc–enabled servers can use Azure Automanage to enroll on-premises systems in the following Azure services:

- Machines Insights Monitoring
- Azure Automation Update Management
- Microsoft antimalware
- Change tracking and inventory
- Azure guest configuration
- Azure Automation accounts
- Azure Log Analytics

Each of these Azure services can be enabled and configured on an individual basis for an IaaS VM or Azure Arc–enabled server. The advantage of Automanage is that it simplifies the process of deploying and configuring each of them for you. You can enable Automanage through the Azure portal or through Azure Policy.

NEED MORE REVIEW? **AZURE AUTOMANAGE**

You can learn more about Azure Automanage at *https://docs.microsoft.com/en-us/azure/automanage/automanage-virtual-machines*.

EXAM TIP

Remember when installing the Azure Monitor Agent to be used by Microsoft Sentinel, you must specify the Azure Log Analytics workspace ID and the workspace key.

Skill 1.4: Secure Windows Server networking

Network traffic carries sensitive information that you don't want intercepted by third parties as it travels across your organizational network or across the internet. Networks are also the primary vector for attackers as they use special tools to probe and penetrate the TCP ports of your organization's infrastructure. In this section, you'll learn how you can protect Windows Server using the built-in firewall feature as well as how to implement connection security rules so that critical servers will only communicate with authenticated hosts.

This section covers how to:

- Manage Windows Defender Firewall
- Implement connection security rules
- Implement domain isolation

Manage Windows Defender Firewall

Windows Defender Firewall with Advanced Security (WDFAS) is a bidirectional, host-level firewall. This means not only can you set it to protect a host from incoming traffic, you can also have it block all or specific outgoing traffic. Generally, you can control outbound traffic only on very sensitive hosts. Controlling outbound traffic minimizes the chance of malware infecting the server that is phoning home. On sensitive hosts, you should allow outbound traffic on a per-application rule. For example, you should create a rule to allow a specific browser to communicate with the internet rather than just allowing outbound traffic on port 80.

WDFAS allows for the creation of rules based on program, port, or predefined rules. There are 45 predefined rules, which include categories such as Remote Desktop, Windows Firewall Remote Management, File and Printer Sharing, and Core Networking. One of the useful things about WDFAS is that it generally enables the appropriate firewall rules when you enable a specific role, role service, or feature. These default rules are appropriate most of the time, but you may wish to edit the properties of these rules to make them more secure.

Firewall profiles

Firewall profiles allow you to treat traffic differently depending on the network to which the computer is connected. For example, the same server might be connected to the following:

- A remote site through a VPN connection that has the private profile applied
- A perimeter network by a network adapter, where the connection has the public profile applied
- To the organization's internal network through a second adapter that has the domain profile applied

The advantage of separate profiles is that each one allows you to have separate sets of firewall rules applied. You might allow Distributed File System (DFS) traffic on one profile, web traffic on another, and SMTP traffic on a third.

Profiles are independent of each other; the settings that you configure for one profile do not apply to other profiles. You can configure the following properties for each profile:

- **Firewall State** You can set this to On (default) or Off.
- **Inbound Connections** You can set this to Block (default), Block All, or Allow. The Block All setting means firewall rules that allow connections are ignored.
- **Outbound Connections** You can set this to Block or Allow (default).
- **Settings** This allows you to configure whether notifications are displayed, whether unicast responses are transmitted to multicast or broadcast traffic, and whether to merge rules when rules apply both locally and through Group Policy.
- **Logging** This allows you to specify logging settings.

Firewall logs are written to the %systemroot%\system32\LogFilres\Firewall\pfirewall.log file. The default settings do not log dropped packets or successful connections, which means that, unless you change the defaults, nothing is logged. Firewall logging is most useful when you've determined that a particular service is unavailable to the network because of the firewall.

You can turn on firewall logging on a per-profile basis by selecting the Customize button next to Logging on each profile's properties page. This displays the Customize Logging Settings dialog box shown in Figure 1-24. You can then enable logging for dropped packets and successful connections. If you are having trouble with a newly installed service on a server, enable the logging of dropped packets to determine the properties of the packets dropped so that you can create a firewall rule that allows your service to be accessed from the network.

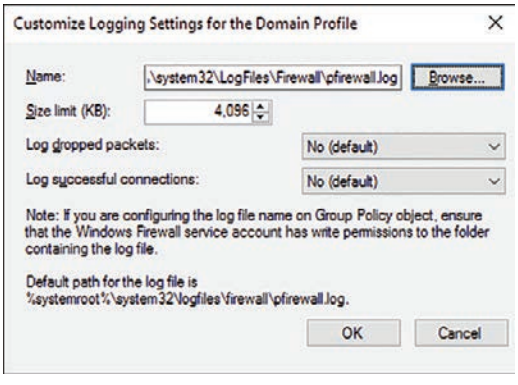

FIGURE 1-24 Custom Logging Settings

Inbound rules

Inbound rules are based on program, port, or one of 45 predefined categories, such as BranchCache Content Retrieval or Network Policy Server. You can also create custom rules

that include a mixture of these, where you specify a program and a port as well as a rule scope. For example, you can use a custom rule to block all incoming traffic on port 80 to a particular application but not block port 80 traffic to other applications on the server. The basic aspects of creating a firewall rule involve:

- **Specifying a program or port** When you specify a port, you must choose whether the rule applies to TCP or UDP traffic.
- **Specifying what action should be taken** You can allow the connection, in which case, all traffic that matches the rule is allowed by the firewall. You can allow the connection if it is authenticated, in which case, traffic that meets the IPsec authentication requirements and the connection security rules is allowed, but traffic that is not properly authenticated according to these conditions is dropped.
- **Specifying the network profiles in which the rule applies** In general, rules should apply in all profiles, but there might be circumstances where you want to allow traffic from an interface connected to a domain network but block the same traffic if it comes from an interface connected to a public network.

After you have created the rule, you can then edit the rule's properties. Editing the rule's properties allows you to configure more advanced options than are present in the Rule Creation Wizard. By editing a rule's properties, you can

- Configure a rule to apply to a service rather than just a program.
- Limit the computers that can make authenticated connections. By default, if you configure the **Allow Traffic If The Connection Is Authenticated** option, any computer that can authenticate is able to successfully transmit traffic. By editing a firewall's rules, you can limit traffic to specific computers, rather than all authenticated computers. You can do the same with user accounts, limiting successful connections to specific users when authentication has successfully occurred.
- Edit the rule's scope, which is the local and remote IP address ranges to which the rule applies. You can also do this when you create a custom rule.
- Configure whether the rule applies to specific network interfaces, instead of just network profiles. For example, if your computer has two network adapters, you can configure a rule to apply so that it allows traffic on one adapter but not the other when both adapters have the same profile set.
- Configure whether or not to allow packets that have passed across a Network Address Translation (NAT) device.

Creating outbound rules

Default settings for Windows Defender Firewall with Advanced Security do not block outbound traffic. In high-security environments, you should consider using outbound rules to block all but authorized outbound traffic.

The process for creating an outbound rule is almost identical to the process for creating an inbound rule. You specify the rule type in the New Outbound Rule Wizard; you specify

whether the connection is blocked, allowed, or allowed only if authenticated; and you specify the network profiles in which the rule is active. By editing the properties of the rule after the rule is created, you can configure all of the advanced options configurable for inbound rules, such as rule scope, specific network interfaces, and to which computers or users the rule allows outbound connections.

Configuring IPsec

The IPsec Settings tab of Firewall Properties allows you to configure how IPsec is used when applied in connection security rules. On the tab itself, shown in Figure 1-25, you can configure whether you want to customize the IPsec defaults, exempt Internet Control Message Protocol (ICMP) traffic from IPsec, and whether you want to configure IPsec tunnel authorization.

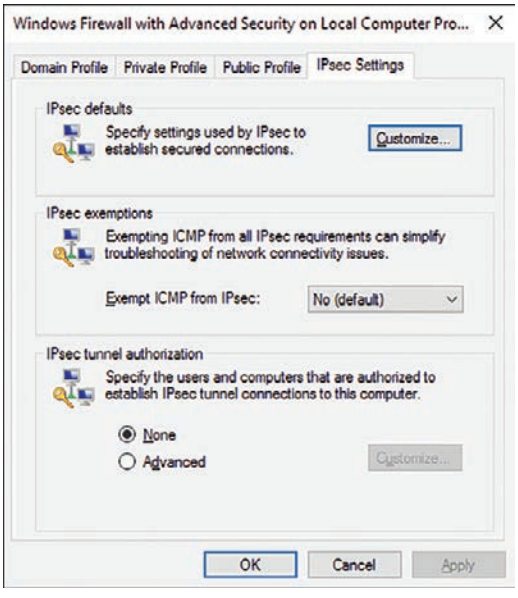

FIGURE 1-25 IPsec Settings

Exempting ICMP traffic can be useful for diagnostic purposes because many administrators use the ping utility to diagnose whether a host has network connectivity. If connection security rules are enabled, the default IPsec settings mean that a successful IPsec negotiation must occur prior to an ICMP response being sent. If the IPsec negotiation is the problem, enabling ICMP response allows you to verify that there is network connectivity and that the problem lies just a bit further up the network stack.

If you select the Customize button next to the IPsec defaults, you can change the Key Exchange, Data Protection, and Authentication Method (see Figure 1-26). The default key exchange uses the Diffie–Hellman Group 2 key exchange algorithm with a key lifetime of 480 minutes. You can modify these settings so that they use more secure methods, but this may reduce the ability to interact with some third-party devices.