**Module 4 – Configuring network connectivity**

**Module Overview**

After you’ve installed Windows, and completed the initial post-installation configuration, you’ll probably need to connect your computer to one or more networks. It’s therefore important that you understand how to configure network settings in Windows 11. You’ll also likely need to know how to create and configure remote access by using a virtual private network (VPN). This module provides guidance on configuring network settings, including how to set up and manage a VPN.

**Objectives**

After completing this module, you will be able to:

* Configure network settings
* Implement name resolution
* Describe remote access options

Lesson 1

**Configuring network connectivity**

It’s interesting to know that the underlying network hardware topology to which most devices are connected, Ethernet, is now almost 50 years old. And the TCP/IP protocol stack is not much younger, having been defined in the 1980s. And despite having been around for a while, these networking standards still provide the backbone of the connectivity systems we use today.

The days of standalone computers without connectivity to other devices are long gone. As soon as you unbox a new Windows 11 computer, the out-of-box-experience (OOBE) wizard prompts you to connect your device to a network so that setup can complete.

From a user perspective, network connectivity can be viewed as a closed system; that’s to say, they can use the network connections on their devices to access apps and services, but they don’t need to understand how they work.

But as an IT Pro, you do need to know how they work. In this lesson, you’ll learn about network connectivity, configuring IPv4 settings, managing name resolution, and enabling remote access.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe the IPv4 protocol
* Describe automatic IPv4 configuration
* Configure Windows 11 network settings
* Troubleshoot network settings
* Implement wireless networking

**Overview of IPv4**

The TCP/IP protocol supports two network transports: TCP and UDP. The stack also provides for two network layer protocols: IPv4 and IPv6. This lesson focuses on IPv4.

Each computer, referred to as a host, on an IPv4 network requires a unique IPv4 configuration. This configuration consists of the following:

* **IPv4 address**. This consists of a network and subnetwork portion (the prefix), and a host portion. All devices in the same subnet should share the same IPv4 prefix, and each host requires a unique host portion within that subnet. For example: 172.16.0.100.
* **Subnet mask**. The subnet mask identifies the number of contiguous bits in the prefix. Often expressed in decimal, but also sometimes expressed using classless notation as a number of bits. For example: 255.255.0.0 in decimal, and /16 in classless notation.
* **Default gateway**. This value identifies the IPv4 address of a router. The router is used when the host wants to communicate with a host in another subnet.
* **Preferred DNS server**. Name resolution is critical in modern networks. The configured Domain Name System (DNS) server is used by an IPv4 host to resolve hostnames into IPv4 addresses.

**Classful addressing**

The Internet Assigned Numbers Authority (IANA) organizes IPv4 addresses into classes, and the number of hosts in a network determines the required class of addresses. Class A through Class E are the names that IANA has specified for IPv4 address classes. Hosts are assigned an address class from A through C. D is used for multicasting, while E is reserved.

The point of classful addressing was to quickly and easily identify the subnet in which a host belonged based on its class. Without getting stuck in the minutiae of binary addressing, you can identify the address class of an IPv4 host as follows. If the first octet of the address falls within the range:

* **1-126**. This is a class A IPv4 host. This means the first octet defines the network while the remaining three octets can be used to identify a host within that network. Class A networks can be very large, containing up to 16,777,214 hosts.
* **128-191**. This is a class B IPv4 host. This means the first two octets are the network. This provides for 16,384 networks. Each of these networks can contain up to 65,534 hosts.
* **192-223**. Identifies a class C network. Theoretically, there can be 2,097,152 class C networks, each containing up to 254 hosts.

This is summarized in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | First octet | Default subnet mask | Number of networks | Number of hosts per network |
| A | 1 to 127 | 255.0.0.0 | 126 | 16,777,214 |
| B | 128 to 191 | 255.255.0.0 | 16,384 | 65,534 |
| C | 192 to 223 | 255.255.255.0 | 2,097,152 | 254 |

**Classless addressing**

The problem with classful addressing is despite being simple, it’s not efficient of the limited address space in IPv4. With a 32-bit binary scheme, the address space supports a maximum of about 4.3 billion addresses. Back in the 1980s, that probably seemed like a lot. But now, that’s nowhere near enough for the internet’s needs. However, as soon as you use the classful addressing scheme, that reduces to around 3.4 billion due to inefficiencies in the way the scheme works.

Classless addressing, sometimes called Classless Inter-Domain Routing (CIDR) notation, enables you to resolve this problem. In CIDR networks, you can use as many subnets bits as you want, regardless of the class. This variable length subnet mask (VLSM) enables you to be very specific about your network configuration.

Remember, class A use 8 bits, B use 16 bits, and C use 24 bits.

You express the number of bits you’ve used by declaring it like this:

172.16.0.100/20

In this instance, 20 bits are used for the subnet rather than the default 16 (for this class B address, the default classful mask would be 255.255.0.0, or /16). That means that you’ve configured a network with the address 172.16.0.0/16 which has been subnetted to create 16 subnets, starting 172.16.0.0/20. The next subnet would be 172.16.16.0/20. The one after that would be 172.16.32.0/20 and so on.

The reason you’d do this is that without subnetting using CIDR, you’d need to use 16 class B networks (in this example). With CIDR you can use only one network, and then subdivide as your needs dictate.

You don’t need to just use 20 bits. You can use more or less. More subnet bits means fewer hosts per subnet, but more subnets.

**Private and public IPv4 addresses**

These days, most computers are not connected directly to the internet. Instead, they connect through routers, firewalls, and network address translation (NAT) devices. Back in the 1990s, it became clear that the pool of available IPv4 addresses was quickly diminishing. After all, the IPv4 address space supports only three to four billion unique addresses.

The IANA came up with a scheme. They decided that for devices sitting in a private (organizationally owned and managed) network, non-routable addresses would be used. On the edge, that’s the bit of the network that connects your organization to the internet, a device would translate these non-routable (or private) addresses to routable (or public) addresses.

The next step was to identify which addresses would be private. The following table lists the available ranges.

|  |  |  |
| --- | --- | --- |
| Class | Mask | Range |
| A | 10.0.0.0/8 | 10.0.0.0 - 10.255.255.255 |
| B | 172.16.0.0/12 | 172.16.0.0 - 172.31.255.255 |
| C | 192.168.0.0/16 | 192.168.0.0 - 192.168.255.255 |

This is why most users in home networks use an IPv4 address that starts 192.168.x.y/16. This rather clever idea has extended the life of the IPv4 protocol by enabling organizations to use only a small number of public IPv4 addresses for internet-facing devices. In the case of home users, a single public IPv4 address is assigned by your service provider to the public side of your network hub.

**Automatic IPv4 configuration**

Although it’s possible, and occasionally necessary, to manually configure the IPv4 settings in your computer, for the most part, it’s simpler and more reliable to use automatic IPv4 configuration. IPv4 supports a system known as stateful autoconfiguration, where a network services allocates and records IPv4 configurations. In a Microsoft environment, the Dynamic Host Configuration Protocol (DHCP) server role performs these tasks.

By default, Windows 11 computers are configured to obtain their IPv4 configurations dynamically via DHCP. This configuration consists of the following:

* IPv4 address
* Subnet mask
* Default gateway
* Preferred DNS server
* DNS suffix

The significant benefit of using DHCP to automatically manage IP addressing in your organization is that it helps reduce IP addressing and configuration errors. In addition, it enables you to centrally manage and monitor your organization’s address space.

Because of the critical nature of the DHCP server role, it’s usual to configure at least two DHCP servers in your organization, and configure a failover configuration between them. This helps minimize disruption if one DHCP server fails.

Many devices, such as wireless hubs, mimic DHCP and are able to allocate IPv4 configurations in the same way that the DHCP server role does. Make sure you disable this functionality when you are using DHCP servers as conflicts might occur.

**How it works**

When a computer starts up, it attempts to obtain an IPv4 configuration. Because it doesn’t initially have an IPv4 configuration, it must a generic means of communicating known as a broadcast. The following process occurs when a computer first starts up:

1. The client device broadcasts a DHCPDISCOVER packet onto the local network. This occurs using UDP port 67.
2. One (or more) DHCP server(s) respond by broadcasting a DHCPOFFER packet that contains a suggested (offered) IPv4 configuration. This occurs using UDP port 68.
3. The client device broadcasts a DHCPREQUEST packet identifying the preferred DHCP server. All DHCP servers that initially responded should receive this packet, again, over UDP port 67. This packet requests the use of the offered configuration (known as leasing).
4. The petitioned DHCP server responds with a broadcast over UDP port 68 to confirm (or occasionally deny) the lease of the configuration.
5. The client now initializes the IPv4 protocol stack using the offered configuration.
6. The DHCP server records the lease details.

The client device continues to use the leased IPv4 configuration until it expires, or until a DHCP server denies the renewal of the lease, whichever is sooner.

The default lease period is eight days.

When the lease approaches the halfway point, the client will attempt to renew. It does this as follows:

1. The client device send a unicast DHCPREQUEST packet to their specific DHCP server over UDP port 67. This packet requests renewal.
2. The petitioned DHCP server responds with a unicast over UDP port 68 to confirm (or occasionally deny) the renewal of the leased configuration.

If the DHCP server doesn’t respond, the client continues using the leased configuration. Client devices also renew during startup, even if the lease has a long time to run. This is largely because client devices often startup in different subnets. The renewal process is as above.

However, if the client device has, in fact, started up in a new subnet, the DHCP won’t respond. In this circumstance, the client sends a packet to its configured router using local delivery. If the client is in a new subnet, the router won’t – in fact can’t – respond. The client then starts the DHCP discovery phase all over again, hoping to contact a new DHCP server more appropriate to their new subnet location.

If a client device cannot obtain an IP configuration, it uses an IPv4 address in the range 169.254.0.0/16. This is known as Automatic IP Address Allocation (APIPA).

**Configuring network settings**

You can configure the network settings in Windows 11 by using the Settings app, Control Panel, Windows PowerShell, and some legacy Command Prompt tools.

**Network & internet**

This is the preferred interface for graphically managing and configuring network settings. From within Settings, select Network & internet.

From the Network & internet page, you can configure the following:

* **WiFi**. Enables you to review and configure your wireless network settings. This includes enabling or disabling WiFi, reviewing available networks, managing known networks, and enabling a random hardware address (for security purposes). You can also manage the hardware properties for your wireless adapter.
* **Ethernet**. Enables you to review and configure your wired network connection. This includes enabling 802.1x authentication, and configuring your IP settings and DNS settings.

Depending on your computer, you might have either or both WiFi and Ethernet.

* **VPN**. Enables you to add, connect, and manage your VPN connections.
* **Mobile hotspot**. Enables you to turn on/off and configure the settings for your mobile hotspot, making your device act as a WiFi hub for other local devices.

You can also configure Flight mode, Proxy settings, Dial-up connectivity, and access Advanced network settings.

**Network and Sharing Center**

To access the Network and Sharing Center, from **Control Panel**, click **Network and Internet**, and then click **Network and Sharing Center**. This was the standard way of reviewing and configuring network settings in earlier versions of Windows, such as Windows 7. It’s hardly changed since then.

Using Network and Sharing Center, you can:

* Review current network connections and settings
* Change adapter settings
* Change advanced sharing settings
* Review and configure Internet Options
* Access the Windows Defender Firewall
* Set up a new connection or network, such as a VPN
* Run the Network and Internet Troubleshooting Wizard

**Using Windows PowerShell cmdlets**

Although you can use graphical tools to perform network configuration and management tasks, Windows PowerShell provides a number of network-specific cmdlets. You can use these to configure, manage, and troubleshoot Windows network connections.

The following table lists some of the network-related Windows PowerShell cmdlets and their purposes.

|  |  |
| --- | --- |
| Cmdlet | Purpose |
| Get-NetIPAddress | Retrieves and displays the IP address configuration. |
| Get‑NetIPv4Protocol | Retrieves and displays the IPv4 protocol configuration (the cmdlet Get-NetIP6Protocol returns the same information for the IPv6 protocol). |
| Get‑NetIPInterface | Retrieves and displays a list of interfaces and their configurations. This doesn’t include IPv4 configuration of the interface. |
| Set‑NetIPAddress | Sets information about the IP address configuration. |
| Set‑NetIPv4Protocol | Sets information about the IPv4 protocol configuration (the cmdlet Set-NetIP6Protocol sets the same information for the IPv6 protocol.) |
| Set‑NetIPInterface | Modifies IP interface properties. |
| Get‑NetRoute | Obtains the list of routes in the local routing table, similar to the route print command. |
| Test-Connection | Runs similar connectivity tests to that used by the Ping command. For example, Test-Connection Adatum-dc1. |
| Resolve-Dnsname | Provides a similar function to the NSLookup tool. |
| Get‑NetConnectionProfile | Obtains the network location profile type to which a network adapter is connected. |
| Clear-DnsClientCache | Purges the DNS resolver cache, similar to the IPConfig /flushdns command. |
| Get-DnsClient | Displays configuration details specific to the different network interfaces on a specified computer. |
| Get-DnsClientCache | Displays the contents of the local DNS client cache, similar to the IPConfig /displaydns command. |
| Get-DnsClientGlobalSetting | Retrieves global DNS client settings, such as the suffix search list. |
| Get-DnsClientServerAddress | Retrieves one or more DNS server IP addresses associated with the interfaces on the computer. |
| Register-DnsClient | Registers all of the IP addresses on the computer onto the configured DNS server. |
| Set-DnsClient | Sets the interface-specific DNS client configurations on the computer. |
| Set-DnsClientGlobalSetting | Configures global DNS client settings, such as the suffix search list. |
| Set-DnsClientServerAddress | Configures one or more DNS server IP addresses associated with the interfaces on the computer. |

For example, to configure the IPv4 settings for a network connection by using Windows PowerShell, use the following cmdlet:

Set-NetIPAddress –InterfaceAlias Wi-Fi –IPAddress 172.16.16.1

**Netsh command line tool**

You also can use the Netsh command‑line tool to configure network settings. For example, to configure IPv4 by using Netsh, you can use the following example:

Netsh interface ipv4 set address name="Local Area Connection" source=static addr=172.16.16.3 mask=255.255.255.0 gateway=172.16.16.1

**Troubleshooting network settings**

Generally, the network components in Windows 11 are pretty reliable. But occasionally, you might need to review and test network connectivity. You can use the following tools to help:

* Event Viewer
* IPConfig
* Ping
* Tracert
* Pathping
* NSLookup

**Event Viewer**

It’s always worth reviewing the logs in the Event Viewer for network-related problems. Event Viewer has four core logs: Application, Security, Setup, and System. There are also operational logs under the Applications and Services Logs heading.

**IPConfig**

Use IPConfig to display the current TCP/IP network configuration. You can also use IPConfig to refresh both DHCP and DNS settings. For example, you might need to flush the DNS cache. The following table provides a brief description of some of the IPConfig command switches.

|  |  |
| --- | --- |
| Command | Description |
| ipconfig /all | View detailed configuration information. |
| ipconfig /release | Release the leased configuration back to the DHCP server. |
| ipconfig /renew | Renew the leased configuration. |
| ipconfig /displaydns | View the DNS resolver cache entries. |
| ipconfig /flushdns | Purge the DNS resolver cache. |
| ipconfig /registerdns | Register/update the client’s host name with the DNS server. |

The Windows PowerShell **Get-NetIPAddress** cmdlet provides similar output.

**Ping**

Use this tools with caution as often, the protocol used by it is blocked by firewalls. However, it can sometimes be helpful in diagnosing basic connectivity issues on your local network. The command sends and receives ICMP echo request messages to the target host and displays the response.

You can also use the **Test-Connection** cmdlet in Windows PowerShell to perform a similar diagnostic.

**Tracert**

The Tracert tool determines the path taken to a destination computer by sending ICMP echo requests. You can then review details of the path taken, and hopefully be able to identify communications issues along the route.

**Pathping**

Similar to the Tracert tool, but Pathping provides more detailed statistics on the individual steps, or hops, through the network.

**NSLookup**

The NSLookup tool displays information that you can use to diagnose the DNS infrastructure. You can use the tool to confirm connection to the DNS server, in addition to the existence of the required records.

The Resolve-Dnsname cmdlet displays similar information.

**Wireless networking**

Wireless networking is ubiquitous. But it’s important you understand how to review and configure the fundamental WiFi settings in Windows 11.

You can access your WiFi settings in Network & internet in the Settings app. As mentioned earlier, you can then review and update your current settings.

There are two main areas you should be familiar with. These are the 802.11x wireless standards, and wireless security standards.

**802.11x wireless standards**

The following table describes the common WiFi standards and their evolution. The dominant standard in new equipment at the time of writing is Wi-Fi 6.

|  |  |
| --- | --- |
| Specification | Description |
| 802.11a | This is the first extension to the original 802.11 specification. It provides up to 54 megabits per second (mbps) and operates in the 5 gigahertz (GHz) range. It is not compatible with 802.11b. |
| 802.11b | This specification provides 11 mbps and operates in the 2.4 GHz range. |
| 802.11e | This specification defines Quality of Service and multimedia support. |
| 802.11g | This specification is for transmission over short distances at speeds up to 54 mbps. It is backward compatible with 802.11b, and operates in the 2.4 GHz range. |
| 802.11n | This specification adds multiple-input and multiple-output, thereby providing increased data throughput at speeds up to 100 mbps. It vastly improves speed over previous specifications, and it supports both 2.4 GHz and 5 GHz ranges. Also known as Wi-Fi 4. |
| 802.11ac | This specification builds on 802.11n to attain data rates of 433 mbps. 802.11ac operates only in the 5 GHz frequency range. Also known as Wi-Fi 5. |
| 802.11ax | The successor to 802.11ac and provides for higher throughout than its predecessor. Also known as Wi-Fi 6. |

**Wireless security standards**

Wired networks can appear more secure than wireless networks. Perhaps this is because a malicious person requires physical access to your network to connect to your wired resources. But that’s not true with a wireless network. Consequently, it’s important you understand the available wireless networking authentication protocols. These are described in the following table.

|  |  |  |
| --- | --- | --- |
| Security type | Authentication | Encryption |
| Open | No authentication (open) | No encryption |
| Shared (not recommended) | No authentication (open) | Shared key |
| WPA-Personal | Plain text passphrase | WPA with a preshared key or pass phrase  Temporal Key Integrity Protocol or Advanced Encryption Standard (AES) |
| WPA-Enterprise | Institute of Electrical and Electronics Engineers, Inc. (IEEE) 802.1x authentication | WPA, Temporal Key Integrity Protocol, or AES |
| WPA2-Personal | Plain test passphrase | WPA2 with a preshared key,  Temporal Key Integrity Protocol, or AES |
| WPA2-Enterprise | IEEE 802.1x authentication | WPA2, Temporal Key Integrity Protocol, or AES |
| 802.1x | IEEE 802.1x authentication | WEP or Dynamic WEP |

**Configuring a wireless connection**

Use the following procedure in Windows 11 to connect to a wireless network:

1. Right-click the network symbol in taskbar corner overflow and select **Network and Internet settings**.
2. If necessary, enable WiFi.
3. In **Settings**, click the **WiFi** tile.
4. Click **Show available networks**.
5. Select the network to which you want to connect and then click **Connect**.
6. When prompted, enter the network security key and click **Next**.

To review wireless network settings, use the following procedure:

1. Open **Settings** and select **Network & internet**.
2. Click the **WiFi** tile.
3. Click **Manage known networks**.
4. Click the desired WiFi connection and review or change the following settings:

* Connect automatically when in range (defaults to on)
* Network profile type (defaults to public)
* Random hardware addresses (defaults to off)
* IP assignment (defaults to DHCP)
* DNS server assignment (defaults to DHCP)

You can also use the netsh command to review wireless connections:

netsh wlan show interfaces

You can also use the following command to review specific profile details:

netsh wlan show profile *name*

**Demonstration: Configuring network settings**

Lesson 2

**Implementing name resolution**

Working with computer names is easier than working with IP addresses, especially for users. In addition, in most network, dynamic address allocation is used, and so there’s no guarantee that a particular IP address remains associated with a particular host.

Name resolution enables applications on host computers to resolve the name of a host to that host’s currently configured IP address. There are a number of ways to resolve names on IP networks, including broadcasts, hosts file lookup, and Domain Name System (DNS) resolution.

It’s important that you know how name resolution works in your network, and how to troubleshoot name resolution. This lesson describes available name resolution methods, and how they work, and you’ll learn how to configure and troubleshoot name resolution in Windows 11.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe name resolution.
* Describe DNS.
* Troubleshoot name resolution.

**What is name resolution?**

Name resolution is the process of resolving a computer name or hostname into the corresponding IPv4 or IPv6 address. There are six methods that Windows computers can use for name resolution.

Depending on the configured methods, a Windows 11 computer might employ several methods in sequence to resolve a name.

**Name resolution methods**

Let’s review some of those available name resolution methods.

**Broadcast**

In the past, when computer networks were fairly simple, and often consisted of a single subnet comprising a few dozen hosts, broadcast name resolution was often used. In broadcast name resolution, a computer that wants to resolve a name to an IP address broadcasts a message onto the local subnet. The petitioned computer responds. The problems with broadcast name resolution are that broadcasts do not normally propagate routers, which limits their use to a single subnet. In addition, broadcasts can generate unnecessary workload on computers attached to the local subnet.

**Multicast**

Microsoft introduced support for the Link-Local Multicast Name Resolution (LLMNR) protocol in Windows Vista. Multicasts are more efficient that using broadcasts, and is useful in scenarios where there is no DNS server available. However, this, too, is an older name resolution protocol which has been largely superseded.

**Hosts file**

In the early days of the internet, users running the Unix operating system on their computers configured a text file, called Hosts, with a list of hostnames and their corresponding IP addresses. This works well, but does require that the user maintains the integrity of the entries in the hosts file. As the internet grew, this became impractical. Although the Hosts file still exists in Windows (located in the C:\Windows\System32\Drivers\Etc folder), it’s not especially relevant in name resolution anymore.

Microsoft introduced a file called LMHOSTS which performs a similar function to Hosts, but is designed to work in Microsoft networks only.

**Domain Name System**

DNS is an industry standard name resolution platform. It’s the method of name resolution employed on the internet. Microsoft supports DNS through the provision of a Windows Server role, and also through managed services provided in Azure. DNS is based on a hierarchical structure and consists of a collection of domains and subdomains in this hierarchy. DNS is the primary name resolution method for Windows-based devices.

**Computer names and hostnames**

Windows computers have two names: a computer name and a hostname.

**Computer names**

Each Windows computer has an assigned computer name. This is a 16-character name where the last character identifies specific services running on the computer. For example, ADATUM-CL22[20h] represents the computer name with the Windows Server service.

By default, Windows setup generates a valid computer name, but you can configure a specific name by provisioning your computer during setup. After installation, you can review or change the computer name in Settings:

1. Select **System**, then select **About**.
2. Click **Rename this PC**.
3. Enter a valid new name, and click **Next**.
4. Click **Restart now** when prompted.

**Hostnames**

A hostname can’t exceed 255 characters in length. It must contain only alphanumeric characters, periods, and hyphens. A hostname can include a DNS suffix, which together create fully qualified domain name (FQDN).

Usually, the Windows computer name is used as the prefix in the FQDN, and the computer’s AD DS domain name is used as the suffix. So, a computer in the Adatum.com domain running Windows 11 named ADATUM-CL22 will have:

* A computer name of ADATUM-CL22
* A hostname of ADATUM-CL22
* An FQDN of ADATUM-CL22.Adatum.com

You can define additional DNS suffixes using the TCP/IP properties of a specific network adapter.

**How name resolution works in Windows**

The precise procedure for name resolution varies based on the particular configuration of your computer, but typically includes the following steps:

1. The computer determines of the petitioned name is the local hostname
2. Next, the computer checks the DNS resolver cache to determine if the name has recently been resolved. If it has, then the cached information is used. It’s important to know that any records in the Hosts file are prepopulated into the DNS resolver cache.
3. Finally, the computer petitions its configured DNS server for the required information.

It’s important to remember that if your computer is configured to use NetBIOS, which is an older session-management protocol, other methods might also be used to resolve the name or names.

After the name is resolved, the information is stored in the DNS resolve cache, which is an in-memory record of recently resolved DNS names and the respective IP addresses. The time-to-live (TTL) value of the resolved records is used to determine how long the records should remain in resolver cache.

**How DNS works**

DNS is a service that resolves hostnames to IP addresses, and also sometimes IP addresses to hostnames.

To manage your organization’s internal DNS namespace, you can implement the DNS Server role on Windows Server. If you’re working in a hybrid environment, with resources hosted in Azure, you can also use Azure DNS, a fully managed DNS service.

It’s typical in your on-premises infrastructure to deploy multiple DNS servers to provide for load balancing and resiliency.

To manage your organization’s external DNS namespace, you’ll need to choose your public DNS domain name, and then register the namespace with a service provider. This is because the internet uses a single DNS namespace.

This public namespace consists of multiple root servers. These root DNS servers have knowledge of the downstream DNS servers in the namespace. These downstream DNS servers, supporting the top-level domains such as .com, .edu, and .gov, are configured so that they can identify the root servers above them, and organizational level DNS servers below in the hierarchy. This arrangement is known as delegation.

**Structure of DNS**

As we already learned, the DNS namespace consists of a hierarchy of domains and subdomains. However, this information must reside somewhere. A DNS zone hosts the records that pertain to a specific part of the namespace. Sometimes a zone maps to a single domain; other times, a zone maps to multiple domains in a parent-child relationship. For example, Sales.Adatum.com and Adatum.com could be either:

* A single DNS zone called Adatum.com that contains a subdomain called Sales.Adatum.com. Name servers exist only for the Adatum.com zone.
* Two DNS zones. Adatum.com is the parent zone, and a delegation exists for the child zone Sales.Adatum.com. Name servers exist for both Adatum.com and Sales.Adatum.com.

DNS uses both forward and reverse lookup zones to satisfy name resolution requests.

**Forward lookup zones**

Forward lookup zones contain records, of which there are several different types. These include:

* **A**. An IPv4 host record, the most common type of DNS record.
* **AAAA**. An IPv6 host record.
* **SRV**. Service records are used to locate domain controllers and global catalog servers.
* **MX**. Mail exchange records are used to locate the mail servers responsible for a domain.
* **CNAME**. Canonical name records (CNAME records) resolve to another host name, also referred to as an alias.

**Reverse lookup zones**

Reverse lookup zones contain PTR records. These are used to resolve IP addresses to host names; this typically occurs in the context of security verification.

**How names are resolved with DNS**

When a computer wants to resolve a name to an IP address or, indeed, resolve another type of record, the application (known as the resolver) petitions the computer’s configured name server for the required record. If the name server has the required information, it returns it to the resolver. But if it doesn’t, it must query another name server in the namespace.

This process is described in detail below for the name www.microsoft.com:

1. A workstation app (resolver) queries the local DNS server for the IP address www.microsoft.com.
2. If the local DNS server does not have the information, it queries a root DNS server for the location of the .com DNS servers.
3. The local DNS server queries a .com DNS server for the location of the microsoft.com DNS servers.
4. The local DNS server queries the microsoft.com DNS server for the IP address of www.microsoft.com.
5. The microsoft.com DNS server returns the IP address of www.microsoft.com to the local DNS server.
6. The local DNS server caches and then returns the result to the workstation.

Caching and forwarding can modify the name resolution process:

* **Caching**. After a local DNS server resolves a DNS name, it caches the results. The record remains in cache for the period defined in the Time to Live (TTL) value for that record. Remember, this is defined by the record owner. Subsequent resolution requests for the same record are satisfied from cache until the TTL expires.
* **Forwarding**. You can configure a DNS server to forward DNS requests to another DNS server. This is useful when you want to avoid referring to root servers. You can define conditional forwarding in which requests that pertain to a specific domain are forwarded to a specific DNS server.

**Configuring DNS settings**

Generally, there’s little to configure on a client computer running Windows 11. This is because the primary DNS server value is configured automatically by using DHCP.

However, you can review the configured name resolution settings by using the **IPConfig /all** Command Prompt tool. The DNS Servers value displays the configured DNS servers for the computer.

You can also use the **Get-NetIPConfiguration** cmdlet in PowerShell. The DNSServer value contains the configured DNS servers for the computer.

**Troubleshooting name resolution**

As you remember, the Windows resolves hostnames using the Hosts file or by using DNS. When you troubleshoot name resolution, you’ll need to know which order these, and potentially other, methods are attempted.

Windows 11 appends the primary and connection-specific suffixes to all names that it is resolving. If the name resolution is unsuccessful initially, Windows 11 applies parent suffixes of the primary DNS suffix. For example, if the DNS resolver attempts to resolve the name ADATUM-CL22, Windows 11 appends the .adatum.com suffix to attempt resolution. If that is unsuccessful, the operating system appends .com to the name, and attempts to resolve it once again. You can configure this behavior from the **Advanced TCP/IP Settings** page.

The primary tools for troubleshooting hostname resolution are **IPConfig** and **NSLookup**, and their Windows PowerShell equivalents **Get-NetIPAddress**, **Get-NetIPv4Protocol**, and **Resolve-dnsname**.

Tip: Ensure that you purge the DNS resolver cache between each resolution attempt.

**The procedure for troubleshooting name resolution**

If you can’t connect to a remote host, and if you suspect a name resolution problem, you can troubleshoot name resolution by using the following procedure:

1. Open an elevated Command Prompt and clear the DNS resolver cache:

IPConfig /flushdns

You can also use the Windows PowerShell cmdlet **Clear-DnsClientCache**.

1. Attempt to verify connectivity to a remote host by using its IP address. If this works, it helps confirm that the problem is name resolution rather than general network connectivity. Be aware that using PING for this activity can generate a false negative because many hosts block the required ICMP echo packets.
2. Attempt to verify connectivity to the remote host by using its host name. For example, run the following PowerShell cmdlet:

Test-connection ADATUM-CL22.adatum.com

1. If the test is successful, the problem is probably not related to name resolution. But if the test is unsuccessful, edit the **C:\windows\system32\drivers\etc\hosts**text file, and then add the appropriate entry to the end of the file. For example, add this line, and then save the file:

172.16.0.51 ADATUM-CL22.adatum.com

1. Run the **Test-connection ADATUM-CL22.adatum.com** command again. This should now be successful.
2. Check the DNS resolver cache to verify that the name resolved correctly. To examine the DNS resolver cache, run the following cmdlet:

Get-DnsClientCache

1. Revert the Hosts file to its initial contents. Purge the resolver cache again.
2. At a command prompt, run the following command, and then review resolver.txt file to identify the failed stage in name resolution:

NSLookup.exe –d2 ADATUM-CL22.adatum.com. > resolver.txt

Alternatively, the Windows PowerShell equivalent command is:

Resolve-dnsname ADATUM-CL22.adatum.com. > resolver.txt

**Interpreting NSLookup output**

You should understand how to interpret the NSLookup command output so that you can identify whether the name resolution problem exists with the:

* Client computer’s configuration
* Name server
* Configuration of records within the name server-zone database

For example, in the following output sample, the client petitions the DNS server for the required information. This is returned for both Pv4 and IPv6. The following output displays the question and answer portions of a typical query with NSLookup.

SendRequest(), len 51

HEADER:

opcode = QUERY, id = 2, rcode = NOERROR

header flags: query, want recursion

questions = 1, answers = 0, authority records = 0, additional = 0

QUESTIONS:

ADATUM-CL22.adatum.com.Adatum.com, type = A, class = IN

------------

Got answer (116 bytes):

HEADER:

opcode = QUERY, id = 2, rcode = NXDOMAIN

header flags: response, auth. answer, want recursion, recursion avail.

questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:

ADATUM-CL22.adatum.com.Adatum.com, type = A, class = IN

AUTHORITY RECORDS:

-> adatum.com

type = SOA, class = IN, dlen = 43

ttl = 3600 (1 hour)

primary name server = lon-dc1.adatum.com

responsible mail addr = hostmaster.adatum.com

serial = 40

refresh = 900 (15 mins)

retry = 600 (10 mins)

expire = 86400 (1 day)

default TTL = 3600 (1 hour)

------------

SendRequest(), len 40

HEADER:

opcode = QUERY, id = 4, rcode = NOERROR

header flags: query, want recursion

questions = 1, answers = 0, authority records = 0, additional = 0

QUESTIONS:

ADATUM-CL22.adatum.com, type = A, class = IN

------------

Got answer (56 bytes):

HEADER:

opcode = QUERY, id = 4, rcode = NOERROR

header flags: response, auth. answer, want recursion, recursion avail.

questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:

ADATUM-CL22.adatum.com, type = A, class = IN

ANSWERS:

-> ADATUM-CL22.adatum.com

type = A, class = IN, dlen = 4

internet address = 172.16.0.39

ttl = 3600 (1 hour)

------------

Name: ADATUM-CL22.adatum.com

Address: 172.16.0.39

This output has been edited for clarity.

**Demonstration: Configuring and testing name resolution**

Lesson 3

**Implementing remote access**

Users increasingly want to work away from the office, perhaps from home. As an IT support professional, it’s important that you understand the available connectivity options for facilitating remote access. This lesson describes the remote access options available in Windows 11. It also describes how virtual private networks (VPNs) work, and explains how you can configure and use a VPN.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe remote access options.
* Describe a VPN.
* Configure a VPN.

**Overview of remote access options**

A typical network infrastructure contains a number of elements to enable and facilitate secure remote access. These include:

* Firewalls and security zones
* Web Application Proxies
* VPN and DirectAccess servers

**What are firewalls and security zones?**

To help manage the flow or network traffic, you might implement firewalls within your infrastructure to create security zones. For example, you might configure dual firewalls that face the internet. This creates three security zones: external, perimeter, and internal. Let’s examine this more closely.

**Firewalls**

A firewall is a security solution. It establishes a barrier between two or more networks. Firewalls are used to manage the flow of traffic between networks by allowing or blocking network traffic based on that traffic’s properties, such as source and destination address, source and destination TCP or UDP port, and packet contents.

Network administrators often place firewalls:

* At a network perimeter
* Between an organization’s intranet and the internet
* Between the intranet and the internal network

This approach creates security zones.

**Security zones**

Security zones help you control the flow of network traffic between devices within an organization’s network infrastructure. For example, suppose you want to allow guest access to the internet from your corporate WiFi network? You’d probably create a screened network that had access only to the internet and not to any internal organizational resources or apps. This is an example of security zones in action.

Most organizations, even small organizations, implement an internal network zone (intranet), a perimeter network zone, and the external (internet) zone.

The perimeter network is the network between an external and an internal firewall. It’s here that you place servers and services that need to be accessible from the internet. For example, you might place a VPN server in the perimeter for your work from home users to access.

The network perimeter approach has the following benefits:

* If a host on the perimeter network, for example a web server, is compromised by a malicious person, a firewall blocks the access to hosts on the internal network.
* Services can be made available to the internet in a protected manner without exposing hosts on the internal network.
* Direct communication between hosts on the internet and hosts on the internal network (and vice-versa) is blocked. This makes it very difficult for a malicious person to access hosts on the internal network, because traffic flow is restricted.

**Perimeter network server roles and hosts**

You typically deploy the following server roles on perimeter networks:

* **External web server**. Should only contain content that you want to make available to the public. Sensitive information should reside only on servers hosted in the intranet zone.
* **Web proxy server**. Clients on the internal network use it to access web-related content on the internet.
* **Reverse web proxy**. Use this server role to publish content from intranet servers without making that content directly available in the perimeter. Typical examples include making users’ mailboxes available across the internet from mailbox servers running in the intranet.
* **SMTP relay**. Routes mail traffic into and out of the organization.
* **DNS forwarder**. Forwards DNS requests from intranet DNS servers to DNS servers on the internet.
* **VPN server**. Enables your users’ devices to connect from the internet to your organization over a VPN.

**What is a Web Application Proxy?**

You might want to provide access to web applications that are on the intranet to users who are WFH across the internet. The process of configuring an app so that it’s accessible from the internet is known as *publishing*.

Windows Server provides the Web Application Proxy role service, which you can use for publishing applications. It functions as a reverse web proxy to provide access to internal corporate web apps to users who remotely connect to the your internal network.

**What is a VPN?**

A VPN provides a connection between components of a private network, through a public network, such as the internet. Tunneling protocols enable a VPN client to establish and maintain a connection to a virtual port that is listening on a VPN server.

There are two types of VPN connections:

* Remote access VPN. Users that work from home, at a customer site, or from a public wireless-access point can use remote access VPN connections to access a server on their organization’s private network.
* Site-to-site VPN. Also known as router-to-router VPN connections. These enable you to have routed connections between separate offices, or between one office and another organization over a public network.

**What is DirectAccess?**

DirectAccess is feature in Windows 11 that enables seamless remote access to intranet resources without first establishing a user-initiated VPN connection. Unlike VPNs that require user intervention to initiate, DirectAccess enables any app that supports IPv6 on a client computer to have complete access automatically to intranet resources. DirectAccess tunneling protocols include: ISATAP, 6to4, Teredo, and IP-HTTPS.

DirectAccess only supports Windows 7, Windows 10, and Windows 11 client computer operating systems. Furthermore, all devices must be AD DS domain-joined member computers. This is because DirectAccess is configured through Group Policy, and non-member computers are unaffected by Group Policy settings.

**What is a VPN?**

Windows 11 supports VPNs and includes a built-in VPN provider. VPN connections have the following properties:

* **Encapsulation**. When you use VPN technology, it encapsulates private data with a header that contains routing information. This information allows the data to traverse the transit (public) network.
* **Authentication**. In order to ensure that communicating parties can communicate securely, they must authenticate. Windows VPN provides numerous methods for authentication.
* **Encryption**. To ensure the confidentiality of your data as it traverses the public transit network, the sender encrypts the data, and the receiver decrypts it.

**Advanced Features of VPNs in Windows 11**

Windows 11 has several advanced VPN profile features. These work to enhance the VPN experience for Windows 11 users and include VPN Reconnect, Always On, App-triggered VPN, Traffic Filters, and Lock-down VPN.

* **VPN Reconnect**. Enables users to automatically reconnect if their VPN session terminates unexpectedly. For example, if a user is connecting a VPN over a cell network, and their phone disconnects, the VPN session is terminated. With VPN Reconnect, the VPN reestablishes without user intervention when network conditions permit.
* **Always On**. Enables the active VPN profile to connect automatically whenever the following events occur: user sign-in, network change.
* **App-triggered VPN**. Enables VPN to connect automatically when a specified app or set of apps starts. You define the apps by using the Package Family Name for universal Windows Platform apps or a file path for traditional Windows desktop apps.
* **VPN traffic filters**. Enables you to determine what traffic to allow on an organizational network based on policies that you create. Lock-down VPN
* **Lock-down VPN**. If you apply a Lock-down VPN policy to a device, no communication is allowed except via a working VPN connection.

**What Is RADIUS?**

Remote Authentication Dial-In User Service (RADIUS) is an industry-standard authentication protocol. Many vendors, including Microsoft, support this standard. It’s used to manage the exchange of authentication information between elements of a remote-access solution.

In the Microsoft environment, elements of the Network Policy Server (NPS) role can function as a RADIUS server or a RADIUS proxy. Devices such as VPN servers, wireless access points, and 802.1x hubs can function as RADIUS clients. The following table describes the components in RADIUS.

|  |  |
| --- | --- |
| Component | Description |
| RADIUS server | Performs centralized connection authentication, authorization, and accounting for wireless, authenticating switch, and dial-up and VPN connections. When using NPS as a RADIUS server, you configure network access servers (NASs), such as wireless access points and VPN servers, as RADIUS clients in NPS. |
| RADIUS proxy | Routes RADIUS messages between RADIUS clients and RADIUS servers that perform user authentication, authorization, and accounting for the connection attempt. As a RADIUS proxy, NPS is a central switching or routing point through which RADIUS access and accounting messages flow. |
| RADIUS client | RADIUS clients are usually NASs such as wireless access points, 802.1X authenticating switches, and VPN servers. |

**Configuring a VPN**

In order to configure a VPN, you’ll need to define the VPN tunneling protocol and the authentication method.

**VPN tunneling protocols**

Windows 11 supports four VPN tunneling protocols. These are:

* **PPTP**. You can use PPTP for both remote access and site-to-site VPN connections. PPTP is quite an old tunneling protocol and other tunneling protocols are more suited to today’s more security conscious environment.
* **L2TP**. A significant improvement over PPTP. The Microsoft implementation of L2TP relies on IPsec in transport mode for encryption services. The combination of L2TP and IPsec is known as L2TP/IPsec.
* **SSTP**. Uses the HTTPS protocol over TCP port 443 to pass traffic through firewalls and web proxies. This is convenient as port 443 is usually already open. SSTP provides a mechanism to encapsulate PPP traffic over the Secure Sockets Layer (SSL) channel of the HTTPS protocol. Enables support for strong authentication methods, such as Extensible Authentication Protocol-Transport Layer Security (EAP-TLS).
* **IKEv2**. IKEv2 uses the IPsec tunnel mode protocol over UDP port 500. IKEv2 supports mobility, making it a good protocol choice for a mobile workforce. IKEv2-based VPNs enable users to move easily between wireless hotspots, or between wireless and wired connections.

**VPN authentication methods**

Authentication of remote access clients critical. Microsoft VPNs support the following authentication methods:

* **PAP**. Password Authentication Protocol (PAP) is the least secure authentication protocol, because it relies on plaintext passwords. Always try to choose another method of authentication.
* **CHAP**. Challenge Handshake Authentication Protocol (CHAP) is a challenge-response authentication protocol. However, because CHAP requires the use of a reversibly encrypted password, you should consider using another authentication protocol, such as Microsoft CHAP version 2 (MS-CHAP v2).
* **MS-CHAP v2**. This option is a one-way, encrypted password, (also known as mutual authentication) process. This is a good choice for remote access authentication.
* **EAP**. With this method, an arbitrary authentication mechanism authenticates a remote access connection. The remote access client and the authenticator negotiate the authentication scheme to use. EAP is an excellent choice for authentication protocol.
* **Other options**. You can enable two other options when selecting an authentication method:
* **Unauthenticated access**. For testing purposes in a non-production environment only.
* **Machine Certificate for IKEv2**. Select this option if you want to use VPN Reconnect.

**Creating a VPN connection**

It’s fairly straightforward to setup a VPN connection in Windows 11. Use the following procedure:

1. Open the **Settings** app.
2. Select **Network & internet**.
3. Click the **VPN** tile.
4. Click **Add VPN**.
5. In the **Add a VPN connection** dialog box, enter the following information and then click **Save**:

* **VPN provider**. Choose **Windows (built-in)** unless you’ve installed a third-party provider.
* **Connection name**. Enter a name for the connection, such as Adatum Head Office.
* **Server name or address**. Enter the computer name (or IP address) or the target VPN server.
* **VPN type**. Choose from IKEv2, SSTP, L2TP/IPsec, and PPTP. The default is Automatic.
* **Type of sign-in info**. Select how you will sign in. Includes: User name and password, Smart card, and Certificate.
* **User name and password**. These are optional and if needed, you’ll be prompted when you attempt to connect your VPN.

You might need to adjust the advanced properties of the VPN after you’ve created it. To do so, use the following procedure:

1. Open **Control Panel**.
2. Select **Network and Internet**.
3. Select **Network and Sharing Center**.
4. Click **Change adapter settings**.
5. Right-click your VPN connection and choose **Properties**.
6. Select the **Security** tab and configure the following, and then click **OK**:

* Type of VPN
* Advanced settings including IKEv2 mobility
* Data encryption: No encryption, Optional, Require, Maximum strength
* Authentication: EAP or Allow these protocols. You can then choose the appropriate protocols.

It’s important to remember that settings selected on the client must be supported on the VPN server, and supported by any remote access policies defined in RADIUS.

**Demonstration: Configuring a VPN**

**Lab: Configuring network connectivity**

**Question:**In the lab, you configured DNS settings on a client computer. What tool could you use to test DNS from a client computer?

**Module Review and Takeaways**

Review Questions

**Question:**You start your computer, and it can’t communicate on the network. You check the IPv4 address. It displays as 169.254.3.27. What’s the likely problem?

**Question:**Your computer cannot connect to a server. Nobody else seems to be affected. What could you try next?

Tools

The following table lists the tools that this module references.

| **Tool** | **How used** | **Where found** |
| --- | --- | --- |
| NSLookup | * Test DNS name resolution | Command Prompt tool included with Windows 11 |
| IPConfig | * Review IP configuration | Command Prompt tool included with Windows 11 |
| Netsh | * Configure IP and network settings | Command Prompt tool included with Windows 11 |

Common Issues and Troubleshooting Tips

| **Common Issue** | **Troubleshooting Tip** |
| --- | --- |
| You cannot connect to a computer by its name | Check if anyone else is affected, and if not, verify name resolution is working with NSLookup. |
| Computer is not able to communicate on the network | Check the IP address. If it’s 169.254.x.y, then check the DHCP server is online and accessible. |