**Module 7 – Managing apps in Windows 11**

**Module Overview**

Users require apps to complete their work using their computers. In fact, a computer is a platform to run apps, whether those apps be financial analysis tools, or games. It’s important that you know how to make these apps available to your users.

In addition, many users will want to download and install additional apps. The Microsoft Store provides a convenient and safe way to facilitate this. Controlling access to the Store is a key part of app management.

Microsoft’s newest web browser, Microsoft Edge, is the default browser in Windows 11. It’s important you know how to configure this browser, and how to configure synchronization of settings between users’ devices.

**Objectives**

After completing this module, you will be able to:

* Choose how to deploy apps to user devices.
* Configure access to the Microsoft Store.
* Configure Microsoft Edge settings.

Lesson 1

**Deploying apps**

Users work with a variety of types of apps. These include traditional desktop apps, such as Microsoft Office. But they also include apps designed for form factors such as tablets. Increasingly, users are accessing their apps using virtualization technologies. As an IT Pro, you’ll need to know how to deliver apps to your users’ desktop computers.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe the Windows application architecture.
* Describe the various different types of app.
* Explain the ways in which you can deploy apps to users’ computers.

**Windows application architecture**

Back in the 90s, when Windows NT first shipped, the predominant operating system for departmental computing was UNIX. Microsoft engineered their new operating system to support UNIX applications.

This meant that organizations using UNIX apps to support business groups, such as accounting or manufacturing, could continue to use their existing apps, but replace the underlying operating system with Windows.

The architecture of Windows, then and now, supports the notion of *environmental subsystems*. The purpose of these subsystems is not natively support apps from a different OS without the need to virtualize that OS. Windows NT, the distant forebear of Windows 11, supported the following subsystems:

* **The client-server subsystem**. Also known as the *Windows subsystem*, it supports native Windows apps.
* **POSIX**. A UNIX standard of the time, and enabled support for natively running UNIX apps.
* **OS/2**. An IBM operating system of the time.
* **Windows on Windows**. Enabled users to continue to run their apps from earlier versions of Windows and the MS-DOS operating systems.

Things have change in the app space, though. It’s less relevant to need to run POSIX apps, and no one likely needs to run OS/2 or MS-DOS apps. But the Linux and Android operating systems are very prevalent. Windows 11 now includes the following subsystems:

* **Windows**. Supports both desktop apps and Universal Windows Platform (UWP) apps.
* **Linux**. Enables users to run their Linux apps natively using their preferred Linux distribution.
* **Android**. Enables users to install and run a specific subset of Android apps available from the Amazon Appstore.

Consequently, users can run apps designed for Windows, Linux, and Android; all without the need to create and manage virtual machines (VMs) to support those apps.

**Types of app**

We’ve already discussed the ability to run apps designed for other operating systems. But Windows also supports the ability to run apps in a number of other ways.

**Desktop apps**

When you deliver a computer running Windows 11 to a user, you’ll typically install desktop apps on that computer. Desktop apps run in the Windows subsystem, either as 32-bit apps, or more usually now, 64-bit apps.

Windows 11 is a 64-bit operating system, but supports 32-bit apps.

Desktop apps:

* Are installed from an .exe or .msi file.
* Launched from a .exe file located in the **C:\Program Files** folder (or **C:\Program Files (x86)** for 32-bit apps).
* Continue to run in the background when users are using other apps.
* Consume more resources (such as processor, memory, and disk space) than other types of app.

Many desktop apps are built in to Windows, including apps like File Explorer, System Configuration, and many others. It’s also probably true to say that the most widely used Windows apps are Desktop apps; for example, Microsoft Office.

**Universal Windows Platform apps**

When Microsoft introduced Windows 8, they also introduced a new app standard. These apps were referred to as either:

* Store apps, because they were delivered by the then new Windows Store
* Modern apps, because they had a different architecture to Desktop apps

These apps were aimed primarily at touch-centric devices, such as tablets and convertibles. They were designed to use fewer resources, and to suspend in the background – much like apps to on smartphones.

These days, the original Modern app has evolved into the Universal Windows Platform (UWP) app. UWP apps:

* Can run on different multiple hardware platforms, including an Intel PC, a Microsoft HoloLens, an Xbox, or a Windows Phone
* Can be delivered via the Microsoft Store, or by other methods
* Suspend when running in the background
* Consume fewer resources
* Are located in the hidden WindowsApps folder in C:\Program Files.

There are a large number of UWP apps built in to Windows 11, including the Settings app, Mail and Calendar apps, Microsoft Store app, Films & TV, and Groove music.

**RemoteApp apps**

The Remote Desktop Protocol (RDP) enables you to connect over TCP port 3389 to a remote computer. Once connected, you can interact with that computer as if you were sitting interactive with it.

Originally designed to allow users to connect to their office-based desktop computers from home, RDP connectivity has become ubiquitous, providing a means to connect to remote servers and workstations for remote administration.

RDP is also used in Windows 11 Quick Assist and Remote Assistance.

Remote Desktop Services (RDS) is a collection of Windows Server role services that you can use to leverage RDP connectivity. RDS enables you to implement a Virtual Desktop Infrastructure (VDI) to create either:

* VM-based desktop deployments
* Session-based desktop deployments

After RDS components are installed, you can publish apps on the server. This means that:

* The server is installed with the required app
* The server uses its resources to run the app
* Remote users connect over RDP to a running instance of the app

You’ll typically want to use RemoteApp apps in scenarios where:

* Your users’ computers are running a 32-bit version of Windows 10, and the required app is 64-bit only.
* Your users’ computers only have 4 GB of memory, but the app requires 8 GB of memory.
* The app is a legacy app, designed for an older version of Windows, and doesn’t support running directly on Windows 11.
* Your users are using an OS that doesn’t support the app natively, for example, a Linux-based tablet or smartphone.

After your server administrator has installed the required services on your Windows Servers, they’ll published the required apps to a specific URL on your on-premises network. You must then use Control Panel to configure your users’ devices RemoteApp and Desktop Connections settings to point to that URL. After completing this procedure, users can access the published apps as shortcuts from Start.

**App-V apps**

You can use Microsoft Application Virtualization (App-V) to virtualize your apps. By using App-V, your users run their apps locally. However, the apps aren’t installed on the local computer.

The App-V client provides a simulated operating system environment which provides access to specially prepared virtualized apps that run within that simulated environment. These apps don’t interact with the Windows 11 operating system directly, but instead interact with the App-V client (acting as a proxy) installed on Windows 11.

The App-V client is built in to Windows 11 Enterprise but must be enabled using Group Policy or the Windows PowerShell **Enable-Appv** cmdlet.

Consider using App-V in the following situations:

* **Use client resources**. When your client computers have adequate hardware resources, App-V improves performance over running apps in a remote app context.
* **Minimize app conflict**. When you install apps using App-V, there are no app conflicts because all App-V apps runs in their own isolated environments.
* **Run multiple versions of apps**. You can use App-V to run multiple instances of your apps concurrently on the same client computer, even if the instances are based on different versions of the app.
* **Simplify app removal**. Because App-V apps aren’t installed locally, you can remove them completely and easily.
* **Simplify app upgrades**. You can easily replace one version of an app with an updated version of the app with minimal effort.

**Using VMs to run apps**

Many organizations use line of business apps (LOB apps). These apps are critical to the organization’s primary business function. If these apps are designed for earlier versions of Windows, for example Windows XP, they might not run properly, or at all, on Windows 11.

Possible solution to this problem might include rewriting the app so that it works natively with Windows 11. However, that might not always be an option. In those circumstances, you could try running the apps in a VM.

Consider using the Windows 11 Client Hyper-V feature, and developing a VM running Windows XP installed with the required LOB app. Whenever the user requires access to the app, they access it from within the VM. This might not be an optimum solution to the problem, but it’s a potential workaround.

**Deploying apps to Windows 11**

There are numerous ways you can choose to deploy apps to your users’ devices. The method that you use depends on many factors, including the type of the app. The following methods are typically available:

* Manual local installation
* Sideloading
* Inclusion in desktop images
* Microsoft Deployment Toolkit (MDT)
* Configuration Manager
* Group Policy
* Microsoft Intune

Let’s examine these in more detail.

**Performing manual local installation**

If you’ve only got one or two Windows 11 computers to install an app on, then using a manual local installation is a good choice. The app will provide an installation program with an .exe or an .msi file extension.

Desktop apps that install from an .msi file require the Windows Installer service be running.

You can double-click the installation file and follow the on-screen prompts to install the app. If you’ve got an .msi installer, then you can also use the **Msiexec.exe** command-line tool to manage installation. For example, if the installation source files are located on a file server, you could run the following command to install the app.

Msiexec.exe /i \\Adatum-SVR1\apps\Adatum-app1.msi

**Sideloading**

If you have a UWP app available locally, sometimes referred to as an application package, you can sideload the app on your Windows 11 computers. Remember that UWP apps are often deployed from the Microsoft Store. Sideloading enables you to install the app without accessing the store.

You’ll need to make sure that your app supports offline licensing. Check with your software vendor.

Sideloading is prohibited by default as a security protection. You can enable sideloading using the Settings app:

1. Open **Settings**.
2. Select **Privacy & security**.
3. Select **For developers**.
4. Enable the **Install apps from any source, including loose files** setting.
5. In the **Use developer features** dialog box, click **Yes**.

Your UWP app should come as an .appx file. The app should be digitally signed by a trusted Certification Authority (CA). You can now use the **Add-AppxPackage** PowerShell cmdlet to sideload your app.

**Including apps in desktop images**

If your organization uses desktop images to deploy a standard Windows 11 desktop environment, you can include any required apps (both Desktop and UWP) in the images. That way, when you apply a desktop image to a new computer, the app is deployed along with the operating system.

The downside of this approach is that you’ll need to update your apps within your stored images when updates are made available from the software vendor. If you don’t, the apps you’re deploying will be increasingly out of date.

**Using MDT**

MDT is a free, downloadable deployment tool which you can use to deploy Windows operating systems. Using MDT enables you to perform lite-touch deployments. You can integrate MDT with additional deployment and desktop management tools, including Windows Assessment and Deployment Kit (Windows ADK), and Configuration Manager to perform zero-touch deployments.

You can use MDT to deploy desktop images to your new computers, using a wipe and load deployment approach. One of the useful features of MDT is that in addition to being able to deploy images that contain apps (see above), you can also create a task sequence to install an app during the application of an image.

However, after the installation of Windows is complete, you cannot use MDT to add additional apps; MDT is a deployment tool, not a management tool.

**Using Configuration Manager**

Configuration Manager is a complete desktop computer management system. It includes features for Windows deployment, but also for application deployment and management. Configuration Manager supports the automated deployment of desktop apps, UWP apps, and App-V apps (.exe, .msi, .appv, and .appx file formats).

Your computers must have the Configuration Manager client installed to receive software from Configuration Manager.

Configuration Manager provides the following features:

* **Multiple deployment types**. Enables you to configure a single app deployment, but to make it possible to occur in different ways, depending on specified conditions. For example, install an app locally if a user is signed in to their primary device, but stream the app if the user is signed in elsewhere.
* **Wake on LAN**. Enables you to schedule app deployments to occur after normal business hours.
* **Reporting**. Enables you to determine how successful your app deployment is after it completes.
* **Collections**. Defines collections of users or computers, or collections based on queries of user or computer properties. You then can target your app deployment to these collections.
* **Software inventory**. Provides you with a list of deployed apps.
* **Software metering**.Identifies how often particular apps are used.
* **Asset Intelligence**. Enables you to check software-licensing compliance.

**Using Group Policy**

Group Policy deployment is simple to use, but somewhat limited compared with Configuration Manager. Using Group Policy, you can create a GPO, link it to an Organizational Unit (OU), and configure the policy settings to deploy the apps to the targeted users or computers in the OU.

GPO app deployment has the following requirements and properties:

* You can only deploy Windows Installer .msi file format apps.
* Your computers must be Active Directory domain-joined.
* You can apply to either user or computers.
* You can target any container that supports GPO linking: sites, domains, and OUs.
* You can use Windows Management Instrumentation (WMI) filtering and Security Group filtering to further target your app deployments.

GPO app deployment supports two deployment methods:

* **Assign**. You can assign apps to:
* **Users**. The app installs when the user signs in.
* **Computers**. The app installs when the computer starts.
* **Publish**. You can only publish apps to users. Typically, published apps appear in Control Panel Programs for users to choose to install.

GPO app deployment has the following limitations:

* **Scheduling**. Deployment occurs the next time a Group Policy refresh occurs. There’s no way to control this.
* **Prerequisite checking**. There’s no way to verify that targeted devices meet the requirements of the apps you want to deploy. You do have some basic GPO filtering that you could use to help mitigate. For example, a WMI filter could check the hardware resources or operating system details.
* **Reporting**. There is no reporting feature, and so you’ll need to manually check that app deployment occurred successfully. You’ll be able to use generic GPO troubleshooting tools and techniques, but nothing specialized for app deployment.

**Using Microsoft Intune**

Many organizations subscribe to Microsoft 365. Depending on the specific subscription details, this means they can use Microsoft Intune to perform Mobile Device Management (MDM).

Microsoft Intune enables you to deploy, configure, manage, and secure apps for Windows devices, and devices running iOS, Android, and macOS. Using Intune, you can target your apps to Windows devices that are enrolled in Intune.

To deploy apps using Intune, sign in using an Azure Active Directory (Azure AD) account with Global Admin privileges. Navigate to the Microsoft Endpoint Manager Admin Center.

You can then add the following types of apps to your deployment inventory:

* Store apps for Android, iOS, Google Play, and Microsoft.
* Microsoft 365 apps (Office apps) for Windows and macOS.
* Microsoft Edge for Windows and macOS.
* Microsoft Defender for Endpoint for macOS.
* Other apps:
* Web link apps
* Built-in apps
* LOB apps
* Windows apps (Win32)
* macOS app (DMG)
* Android Enterprise system apps

To use Intune, you select the type of app, configure the deployment properties, and then assign the app deployment to a specific user or device group. Intune provides some great inventory and reporting tools, so it’s easy to review and troubleshoot app deployment.

Lesson 2

**Managing access to the Microsoft Store**

Users like to install their own apps on their own devices, but they need a safe way to be able to do so. Stores, managed by Google, Apple, and Microsoft, provide a way to do this. Apps available in a store can be assumed to be safe and to pose little security risk compared with downloading and installing apps from websites.

Many of the built-in apps in Windows 11 are UWP apps delivered from and updated by the Microsoft Store. However, in an organizational context, you might not want your users to have unrestricted access to the Microsoft Store.

In this lesson, you’ll learn about the Microsoft Store, how to configure the Windows 11 Microsoft Store app, and how to restrict access to the Microsoft Store.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe the Microsoft Store.
* Restrict access to the Microsoft Store.
* Maintain UWP apps.

**What is the Microsoft Store?**

Microsoft introduced their online app store, the Windows Store, to support the deployment of apps for users of Windows 8. It’s now referred to as the Microsoft Store, and has evolved since its inception.

Originally, the Windows Store supported the deployment of Modern apps only (now known as UWP apps). However, the Microsoft Store also supports the deployment of desktop apps (Win32 apps).

This makes for a safer way than browsing websites to locate and download desktop apps from third party software vendors. In fact, you can prevent users from downloading and installing apps except from the Store by configuring the **Choose where to get apps** setting in the **Settings** app to use **The Microsoft Store only** value.

**The Microsoft Store App**

In Windows 11, the Microsoft Store is accessed using the Microsoft Store app. The app lists your purchased apps, games, and movies. It also provides the Library, which is the list of installed apps and updates to those apps on your device.

To download and install apps, you must sign in to the Microsoft Store app using an organizational account (Azure AD account associated with a Microsoft 365 license) or a consumer Microsoft account (one associated with Outlook.com or Hotmail.com).

**Microsoft Store for Business**

Organizations that have a Microsoft 365 subscription also have access to the Microsoft Store for Business. This store enables you to purchase and distribute apps for your users. You can also create a custom store front, or private store, that lists only your purchased apps. However, the Microsoft Store for Business is due for retirement in Q1/2023.

**Restricting access to the Microsoft Store**

If you want to prevent your users from accessing the Microsoft Store, you can use Group Policy, Microsoft Intune, or by editing the computers’ registry settings.

To use Group Policy, use the following procedure:

1. Open **Group Policy Management**, and locate the appropriate GPO.
2. Open the GPO for editing.
3. In the **Group Policy Management Editor**, navigate to: **Computer Configuration / Policies / Administrative Templates / Windows Components / Store**.
4. Locate the **Turn off the Store application** setting and double-click it.
5. In the **Turn off the Store application** dialog box, click **Enabled** and click **OK**.
6. Close all open windows.

You can also create an AppLocker policy that prevents the Store app from running.

1. Click **Start**, type **secpol** and then press Enter. The Local Security Policy opens.
2. Expand **Application Control Policies**, expand **AppLocker**, and then click **Packaged app Rules**.
3. Right-click **Packaged app Rules**, and then select **Create New Rule**.
4. On **Before You Begin** page click **Next**.
5. On the **Permissions** page, click **Deny** and then click **Next**. The default setting is to configure the block for the group Everyone. You can, of course, be more specific.
6. On the **Publisher** page, select **Use an installed app package as a reference**, and then click **Select**.
7. In the **Select applications** dialog box, locate and select the **Microsoft Store** app, and then click **OK**.
8. On the **Publisher** page, click **Next**.
9. On the **Exceptions** page, define any exceptions to the rule, and then click **Next**.
10. On the **Name** page, add a meaningful name, and then click **Create**.

You’ll need to ensure the Application Identity service is running on all computers. This service is required for AppLocker.

If your organization uses Intune, you can create a device configuration profile that will block the Store app:

1. In the Microsoft Endpoint Manager admin center, click **Devices** and then click **Configuration** **profiles**.
2. Click **Create profile** and then choose **Windows 10 and later** in the **Platform** list.
3. In the **Profile type** list, choose **Templates**, then choose **Administrative Templates** and click **Create**.
4. In the wizard, on the **Basics** tab, enter a name and description and click **Next**.
5. On the **Configuration settings** tab, under **Computer Configuration**, expand **Windows Components** and locate and expand **Store**.
6. Click the **Turn off the Store application** setting.
7. Click **Enabled** and click **OK**.
8. Click **Next**, and then complete the wizard by assigning the profile to a user or device group that targets your Windows 11 computers.

**Maintaining UWP apps**

As mentioned earlier Windows 11 comes preinstalled with a large number of Store apps. These built-in apps provide core functionality in Windows. You can perform the following UWP app maintenance tasks:

* Manage updates
* Remove apps
* Troubleshoot UWP apps

**Manage UWP app updates**

You can configure updates for these apps by configuring the Microsoft Store app’s settings.

1. Open the **Microsoft Store** app and click **Profile** (the icon for your account in the upper right of the window).
2. To enable automatic app updates, enable the **App updates** setting.

If you want to manually update specific apps, use the following procedure:

1. In the **Microsoft Store** app, click **Library**.
2. Under the **Updates & downloads** heading, review any available updates.
3. To update a specific app, locate the app in the list and then click **Update**.

**Remove UWP apps**

To remove apps, you can locate the app in the Start menu, right-click the app, and choose **Uninstall**. If the app is a UWP app, you are prompted to click **Uninstall** to confirm. If, however, the app you selected is a desktop app, Control Panel opens, and you see the **Uninstall or change a program** page.

You can also remove apps from the Settings app:

1. In **Settings**, click **Apps**.
2. Click **Apps & features**. A list of apps displays, including installed desktop apps.
3. Locate the app you want to remove, and click the ellipsis button at the right of the app’s tile. Click **Uninstall**.

Many built-in apps can’t be removed in this way.

**Troubleshoot UWP apps**

If you experience problems with a UWP app, you can try a few things to resolve the issue with the app. These include:

* Uninstalling the app
* Moving the app from the C: drive to another location (not all apps can be moved)
* Access advanced options:
* Terminate the app. This stops the app enabling you to perform a fresh restart.
* Repair the app. This doesn’t remove app data.
* Reset the app. This removes app data.

To configure advanced options, from the Settings app:

1. In **Settings**, click **Apps**.
2. Click **Apps & features**.
3. Locate the app you want to remove, and click the ellipsis button at the right of the app’s tile. Click **Advanced options**.
4. Details for the app display. Choose between **Terminate**, **Repair**, and **Reset**.

**Demonstration: Maintaining UWP apps**

Lesson 3

**Configuring Microsoft Edge**

The preferred Microsoft web browser for Windows 11 is Microsoft Edge (based on Chromium). This browser was released in January 2020, and comes preinstalled on Windows 11. It’s important that you know how to configure Microsoft Edge so that you can help support your users with this new browser.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Configure Microsoft Edge.
* Sync your settings between devices.
* Manage browser extensions.
* Implement Application Guard.

**Configuring Microsoft Edge**

Microsoft Edge replaces Internet Explorer as the default browser in Windows 11. Indeed, Internet Explorer isn’t included by default. For users of Google Chrome, the transition shouldn’t be too challenging; this browser is based on Chromium.

**Importing favorites**

If you’ve been working with another browser and want to import your favorites, that’s fairly easy. You can import from:

* Microsoft Edge Legacy (the first non-Chromium version of Edge).
* Microsoft Internet Explorer.
* Favorites or bookmarks HTML file.
* Passwords CSV file.

When you import your favorites, you can choose to import the following:

* Favorites or bookmarks.
* Saved passwords.
* Search engines.
* Personal info.
* Payment info.
* Browsing history.
* Cookies.
* Home page.
* Settings.
* Open tabs.
* Extensions.

The specific items you can import vary depending on where you’re importing from.

**Configuring page layout**

You can configure the layout of the default page in Edge. Choose between:

* **Focused**. Just the bare essentials are displayed.
* **Inspirational**. Includes an background image of the day, plus content from My Feed.
* **Informational**. As for inspirational, but the focus is all about My Feed.
* **Custom**. Enables you to choose your own image background, and customize Quick Links, and determine whether My Feed displays.

**Configuring Settings**

There are a huge number of setting that you can configure. These include:

* **Profiles**. Enables you to define the linked account, and define synchronization settings, personal info, passwords, payment info and more that’s associated with the linked account.
* **Appearance**. Choose light or dark mode, and then select a theme. You can also specify details about what appears on the toolbar (Home, Favorites, Extensions and other buttons, for example).
* **Start, home, and new tabs**. Define the default behavior for the start page, the target for the home button, and what happens when a user opens a new tab.
* **Default browser**. Enables you to choose your default browser (assuming several are installed). You can also configure behaviors for Internet Explorer mode, if IE is installed.
* **Phone and other devices**. Enables you to sync your settings and open tabs to your phone running iOS, Android, and devices with Windows and macOS.

You can configure many of these settings using Group Policy.

**Synchronizing your settings**

In order to sync your settings between your devices with Microsoft Edge, you must sign in using an account that supports synchronization. This means using either a Microsoft consumer account ( Hotmail.com or Outlook.com, for example), or using an organizational account (Azure AD).

You enable sync:

1. Open Edge and click the ellipsis button on the toolbar.
2. Click **Settings**.
3. On the **Profiles** tab, click the **Sync** tile.
4. You can then click **Turn on sync** / **Turn off sync** (as appropriate).
5. After you enable sync, you can sync the following:

* Favorites.
* Settings.
* Basic info.
* Passwords.
* History.
* Open tabs.
* Extensions.
* Collections.

If you’re experiencing problems with sync, click the Reset sync button. Otherwise, consider turning off sync, and then turning it on again.

**Managing extensions**

Extensions enable you to modify the default behavior of websites. They can also add functionality to your browsing experience.

Extensions fall into a number of categories, including:

* Accessibility
* Communications
* News and weather
* Ad blocking
* Search tools
* Shopping

To add extensions, open Edge and then:

1. Click the ellipsis on the toolbar.
2. Select **Extensions**.
3. Click **Manage extension**.
4. On the **Installed extensions** page, click **Get extensions for Microsoft Edge**.
5. Select any extensions and follow the on-screen instructions to install and configure them.

If you can’t find what you need, back on the Extensions tab in Edge, you can click the Chrome Web Store link. Edge supports most extensions designed for Google Chrome.

You might need to sign in to the Google Chrome store using a Google account.

**Implementing Application Guard**

One of the biggest security risks faced by your organization is when users follow links in phishing emails. These links are usually to websites with unsafe or malicious content. If users access such sites’ content, then downloaded files can exploit potential weaknesses on their computers, and these exploits can wreak security havoc on your network.

Although educating your users to ignore and report such emails, inevitably, users do act on emails and follow the links they contain. Windows 11 contains many security settings to help mitigate these and other potential threats. Microsoft Edge supports a feature called Microsoft Defender Application Guard.

Microsoft Office also supports Application Guard.

Application Guard uses virtualization-assisted security to isolate the Edge browser session. Because Application Guard isolates the session, even if a user access malicious files and content, that content cannot exploit the host computer.

**Requirements**

To enable Application Guard, your computer must meet the following requirements:

* Windows 11 Enterprise, Education, or Professional
* 8 GB of physical memory
* Support for Virtualization-based security
* UEFI 2.3.1 or greater
* Secure Boot
* TPM 2.0 (either discrete or firmware)
* UEFI (firmware) lock
* Virtualization features: Intel VT-x, AMD-V, and SLAT must be enabled

**Enabling Application Guard**

On a single computer, you can enable Application Guard using the Windows Security app:

1. Open **Windows Security**.
2. Select **App & browser control**.
3. Click the **Install Microsoft Defender Application Guard** link.

You can also enable the feature through Control Panel. Choose Turn Windows features on or off, and then select Microsoft Defender Application Guard.

After you’ve enabled the feature, then in Windows Security, click **Change Application Guard settings**. You can configure the following behaviors:

* **Save data**. Retains any data users save while browsing.
* **Copy and paste**. Allows copy and paste between the browser session and the host.
* **Print files**. Enables printing from the isolated session.
* **Camera and microphone**. Enables access to these devices.
* **Advanced graphics**. Improves graphics performance.

Enabling any of these features increases usability, but at the potential cost of reducing security.

If you need to configure multiple computers, you can use either Group Policy or Microsoft Intune to configure Application Guard settings. The Group Policy node is located at: **Computer Configuration / Policies / Administrative Templates /Windows Components / Microsoft Defender Application Guard**. There are additional configuration settings available via Group Policy.

**Demonstration: Implementing Application Guard**

**Lab: Managing apps**

**Question:**In the lab, you enabled Application Guard. What’s the main benefit of this feature?

**Module Review and Takeaways**

Review Questions

**Question:**You have a legacy LOB app designed for Windows XP. It doesn’t run properly in Windows 11. What options could you consider?

**Question:**You want to install a packaged UWP app. Which PowerShell cmdlet would you use?

**Question:**True or False? You can use MDT to deploy apps?

Tools

The following table lists the tools that this module references.

| **Tool** | **How used** | **Where found** |
| --- | --- | --- |
| Msiexec.exe | * Installs .msi apps | Command Prompt |
| Secpol.msc | * Management Console snap-in for accessing Local Security Policy | Windows Tools |