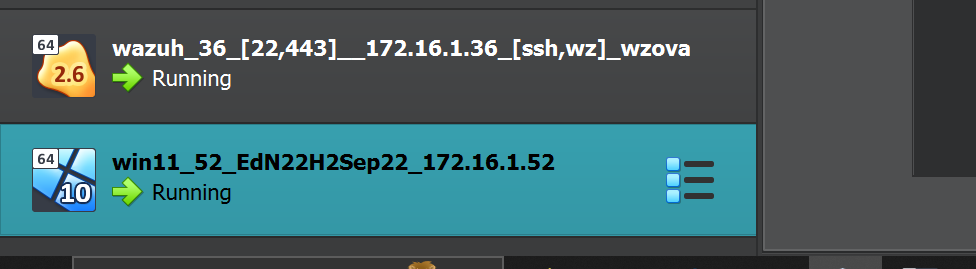
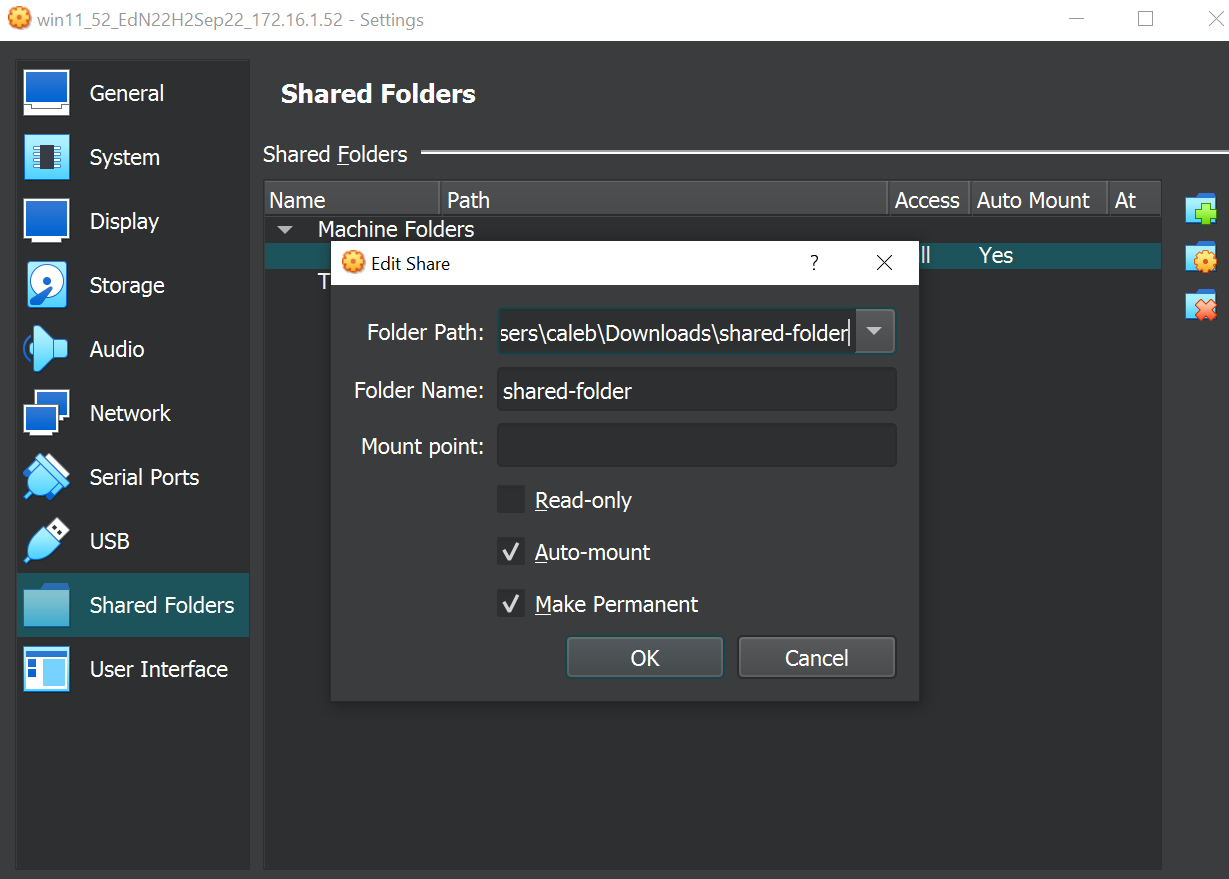
# Topic 2 Implement safeguard CIS 2.5 Allowlist Authorised Software

Your boss uses the Win11 virtual machine. Download the Win11 virtual machine from the uni website and import it into Virtualbox.

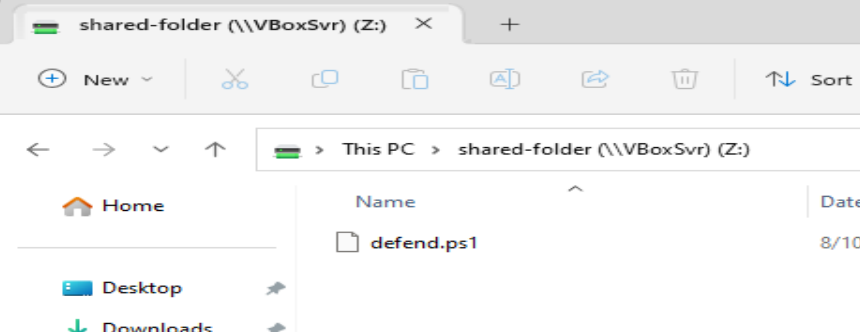


Setup a shared folder on Win11 for your scripts.



To implement an allowlist of authorised software you will configure Windows Defender Application Control (WDAC) in Win11. The following instructions are a guide. You can make improvements as you see fit.

1. Create a PowerShell script called defend.ps1.



Pre-test: write a function testInternetAccess():

a. Use Invoke-WebRequest to check for Internet access by downloading a webpage. An HTTP status code of 200 means that a webpage was successfully obtained. You might need to use the option -UseBasicParsing.

b. The function should interpret the HTTP status and output to the console whether the command was successful or not.

3. Document your code.

a. Clean up your code using PSScriptAnalzyer

PS> Invoke-ScriptAnalyzer defend.ps1

b. Run the code and collect testing screenshots for your portfolio.

c. Commit your code to your Git repository.

4. Create a function enableRestrictInternet() that removes Internet access by setting the proxy to a fake, non-existent server “proxy”: [System.Net.HttpWebRequest]::DefaultWebProxy = ` New-Object System.Net.WebProxy("http://proxy",$true)

5. Add code to your script that checks the first argument of the command line:

a. If the first argument is testInternetAccess, call the testInternetAccess() function. For example, the following command line would call testInternetAccess(): defend.ps1 testInternetAccess

b. Improve your code to call the other functions in your script based on the command line argument.

6. Post-test: After running this function, rerun testInternetAccess(). Internet access should now be disabled due to the fake web proxy. Run the code and collect screenshots for your portfolio.

7. Add a resetRestrictInternet() function to your script that re-enables Internet access by removing the fake proxy: [System.Net.HttpWebRequest]::DefaultWebProxy = ` New-Object System.Net.WebProxy($null)