



Table of contents

Labs

- [Lab 1](#)
- [Lab 2](#)

Assignments

- [Assignment 1](#)
- [Assignment 2](#)

Lab 1 - Environment Setup

Investigation 1: Downloading Installation Media

1. Navigate to the Microsoft Azure site: portal.azure.com
2. Use your Seneca e-mail address and password to login.
3. Once on the main Azure page, look for the Search bar at the top of the page.
4. In the Search bar, type: Education, then hit Enter.
5. In the Education | Overview page, look to the left. You will see menu items already displayed on screen. (Overview, Learning resources, etc.)
6. Inside Learning resources in the left menu, click on Software.
7. In the main Software page, there is a Search bar just below the word Software (it says *Search* inside it.)
8. In that search field, type and enter: Windows Server 2025 Datacenter
9. In the item that appears below (there should only be one), click the link for Windows Server 2025 Datacenter.
10. On the right, an information box appears describing the software. Using your mouse to hover over this information box, scroll down to the bottom.
11. You should now see two items: **View Key** and **Download**
12. Click on **Download** first to begin downloading the Server 2025 ISO. You will need this for your operating system installation. (Don't forget where you've saved it!)
13. While the ISO file is downloading, click on **View Key**.
14. Copy this key into a text file that you save locally on your personal computer or personal USB key. You will need this for the Server installation and for any reinstalls later in the semester. **Do not lose this key and do NOT share it with anyone!**

Investigation 2: VMWare Workstation

The use of VMware Workstation is required for this course. You can either use a Seneca Lab computer (Option 1) or a personal computer (Option 2). If you use an external SSD, you can use both!

WARNING: Seneca Lab computers erase any saved data when restarted from their local drive. Save all work to an external drive or online storage.

Option 1a: Seneca Lab Computers w/VMware Workstation Installed Locally

How do you know if your particular Seneca Lab classroom has VMware Workstation installed?

When you log in, the VMware Workstation icon will be visible on the desktop. If it is not, jump to *Option 1b*.

1. Log in to the Seneca workstation with your Seneca username (don't include @myseneca.ca) and password.
2. Once the desktop has fully loaded, double-click on the **VMware Workstation** icon on the desktop. (If not present, jump to *Option 1b* instructions.)
3. When the **VMware Workstation** application appears on screen, continue to the next Investigation.

Option 1b: Seneca Lab Computers *without* VMware Workstation

Installed

Do not use this option if your lab computer already has VMware Workstation installed. Go back to Option 1a.

1. Log in to the Seneca workstation with your Seneca username (don't include @myseneca.ca) and password.
2. Once the desktop has fully loaded, double-click on the **Seneca MyApps** icon on the desktop.
3. In the browser that launches, log in with your Seneca credentials.
4. Do not ignore dialog boxes! Read them carefully instead of just clicking *Cancel* or randomly.
5. In the search bar in the middle of the screen, type *VMware* and hit **Enter**.
6. One result should show up, **VMware WS Pro**. Click this result.
7. In the new *VMware WS Pro* page, click on the version number (Example: 17.6.1). This will launch the application over the network.
8. Click on allow for any dialog boxes that appear. Remember, **do not ignore dialog boxes!**
9. Be patient. It can take a bit for the background files to transfer over the network.
10. When complete, the **VMware Workstation** application should appear on screen. (You will also see the **VMware Workstation** icon on the desktop. You can launch it from here if you accidentally close the program.)
11. Proceed to the next investigation.

Option 2: Personal Computer

Part 1: Computer Hardware Requirements

To run this course and its tech on a personal computer, it must meet the following minimum system requirements:

1. CPU: 4-core Intel/AMD (2012 or later)
2. RAM: 16 GB
3. Storage: 250 GB SSD free, *reserved for this course* (if external, USB3.0 or higher)
4. Internet connection: High speed, stable

CPU architecture is vital. You cannot use ARM or Snapdragon based computers. This includes all Apple Silicon computers (M1/M2/M3/M4, etc).

Part 2: Registering Your Broadcom Account

Broadcom (formerly VMware) now requires account registration before you can download the VMware Workstation software. Follow the instructions below to do so. You only have to do this once. (If you already have an account using your Seneca e-mail, skip to Step 3.)

1. Visit the Broadcom registration site: <https://profile.broadcom.com/web/registration>
2. Provide your Seneca e-mail address and continue.
3. You will be sent a verification code. Check your Seneca e-mail and enter the code on the page.
4. On the *Complete your Registration* page, enter your information and click **Create Account**.
5. On the *Registered Successfully!* page, you can skip the rest and simply click **I'll do it later**.
6. At the top right of the screen, click on the **Login** button. (No, you aren't logged in automatically. Yes, it's annoying.)
7. Enter your Seneca e-mail address as your username and the password you chose on Step 4 as your credentials.
8. When complete, you should see your name in the top right of the screen instead of *Login*. If not, attempt to log in

again or ask for help.

Part 3: Downloading VMware Workstation

1. Go to this link: <https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Workstation%20Pro&freeDownloads=true>
2. Click on **VMware Workstation Pro 17 for Windows**
3. Click on the highest version available. (17.6.3 as of last update)
4. On the next screen, click on the blue link inside *I agree to the Terms and Conditions* to open a new tab with their terms and conditions. Do not close this window.
5. Back in the previous window/tab that's still open, now click the checkbox for *I agree to the Terms and Conditions*.
6. Click on the little cloud icon next to the name of the download file.
7. Click the **Yes** button when the dialog box pops up.
8. In the *Trade Compliance and Download Conditions* screen, fill in the following school information:
 - i. Address1: **1750 Finch Ave E**
 - ii. City: **North York**
 - iii. State/Province: **Ontario**
 - iv. Country: **Canada**
 - v. Zip/Postal Code: **M2J 2X5**
9. **Finally** click on the cloud icon again to begin the actual download.
10. If you ever need to download the software again, you won't have to go through any of the registration mess. Follow Part 3, Steps 1-3, then Step 9.
11. Direct link: https://downloads2.broadcom.com/?file=VMware-workstation-full-17.6.3-24583834.exe&oid=38988096&id=yvNR4Ur3i4n6lc4Z3pGm4ZNuhcOR9OzqJ5c71zWGQWtt7jgem-_11MXIHeTdyjW73tEfOS8fwGI=&verify=1750391577-zQ%2BZJXDvlnv3EbAMve2m8vmVtcjOe%2Fk1IQv17jA69gA%3D

Part 4: Install VMware Workstation (TBD)

1. Find the downloaded .exe file and run the installer.
2. Follow all prompts and install.
3. (Free for use option?)

Part 5: Launch VMware Workstation (TBD)

1. Open VMware Workstation from your desktop or the Start Menu.
2. When asked, choose the Free option.
3. Proceed to the next Investigation.

Investigation 5: VM1 Installation - Windows Server 2025 Datacenter

Name: srv1-cjohnson30 RAM: 4GB CPU: 2 cores Storage: 80 GB

Part 1: Setup Instructions

1. In the main window, you should see a large + symbol icon titled **Create a New Virtual Machine**. Click it.
2. In the new dialog box, keep *Typical* selected and click the **Next** button.
3. On the next screen, *Guest Operating System Installation*, do the following:
 - i. Select *Installer disc image file (ISO)*:
 - ii. Now click **Browse**.
 - a. Navigate to where you saved your **Windows Server 2025 Datacenter*** downloaded ISO and select it.
 - iii. Once selected, the previous screen should now say "Windows Server 2025 detected. If it doesn't, you haven't selected the right file, or your download was corrupted. Ask for help.
 - iv. Click **Next**.
4. On the "Easy Install Information" screen, do the following:
 - i. Paste in your serial key.
 - ii. Version of Windows to install: Select *Windows Server 2025 Datacenter*
 - iii. Personalize Windows:
 - a. Full Name: Administrator
 - b. Password (both fields): Select a strong password **you will remember**. You will use this same password for all VMs in this course.
 - iv. **Do not select "Log on automatically"**.
 - v. Click **Next**.
5. On the "Name the Virtual Machine" screen, do the following:
 - i. Virtual machine name: *srv1-senecausername*
 - a. For example, if my Seneca e-mail address is cjohnson30@myseneca.ca, then my Seneca username is *cjohnson30*. This would give me a VM name of *srv1-cjohnson30*.
 - ii. Location: If using an external SSD (like with our lab computers), click **Browse** and navigate to your external SSD.
 - a. Create the following directory structure in your SSD: *OSM620 > Virtual Machines/srv1-cjohnson30*
 - b. Select this new *srv1-cjohnson30* folder.
 - c. Make sure you now see this change in the Location field. (Example: *Z:/OSM620/Virtual Machines/srv1-cjohnson30*)
 - iii. Click **Next**.
6. On the "Specify Disk Capacity" screen, do the following:
 - i. Maximum disk size (GB): **80**
 - ii. Select *Split virtual disk into multiple files*.
 - iii. Click **Next**.
7. On the "Ready to Create Virtual Machine" screen, do the following:
 - i. Click on **Customize hardware...**
8. On the new "Hardware" screen, do the following:
 - i. Select *Memory*, and change the value to: **4096**
 - ii. Select *Processors*, and change:
 - a. Number of processors: **2**
 - b. Number of cores per processor: **1**
 - c. Virtualize Intel VT-x/EPT or AMD-V/RVI: **Checked**
 - d. Virtualize CPU performance counters: **Unchecked**
 - e. Virtualize IOMMU (IO memory management unit): **Checked**

- iii. Select *Network Adapter* and confirm:
 - a. *Connected at power on*: **Checked**
 - b. *NAT*: **Checked**
- iv. Click **Close**.
9. Back in the "Ready to Create Virtual Machine" screen, click **Finish**.
10. The virtual machine should launch.
11. If you get a dialog box about *Side channel mitigations*, check the box for *Do not show this hint again* and click **OK**.
12. Click **Close**.
13. Click on the big **Play** to turn on the virtual machine begin the OS installation. This may take some time.

Part 2: Post-Installation Tasks

- Time Zone
- Name Change
- Windows Activation
- Updates
- Internet Connectivity Check w/IE
- Download and Install Firefox

Investigation 6: VM2 Installation - Windows Server 2025 Core

Name: srv2-cjohnson30 RAM: 2GB CPU: 2 cores Storage: 36 GB

Part 1: Setup Instructions

1. In the main window, you should see a large + symbol icon titled **Create a New Virtual Machine**. Click it.
2. In the new dialog box, keep *Typical* selected and click the **Next** button.
3. On the next screen, *Guest Operating System Installation*, do the following:
 - i. Select *Installer disc image file (ISO)*:
 - ii. Now click **Browse**.
 - a. Navigate to where you saved your **Windows Server 2025 Datacenter** downloaded ISO and select it.
 - iii. Once selected, the previous screen should now say "*Windows Server 2025 detected*". If it doesn't, you haven't selected the right file, or your download was corrupted. Ask for help.
 - iv. Click **Next**.
4. On the "Easy Install Information" screen, do the following:
 - i. Paste in your serial key.
 - ii. Version of Windows to install: Select *Windows Server 2025 Datacenter (Core)*
 - iii. Personalize Windows:
 - a. Full Name: Administrator
 - b. Password (both fields): Select a strong password **you will remember**. You will use this same password for all VMs in this course.
 - iv. **Do not select "Log on automatically"**.
 - v. Click **Next**.

5. On the "Name the Virtual Machine" screen, do the following:
 - i. Virtual machine name: *srv2-senecausername*
 - a. For example, if my Seneca e-mail address is *cjohnson30@myseneca.ca*, then my Seneca username is *cjohnson30*. This would give me a VM name of *srv2-cjohnson30*.
 - ii. Location: If using an external SSD (like with our lab computers), click **Browse** and navigate to your external SSD.
 - a. Create the following directory structure in your SSD: *OSM620 > Virtual Machines > srv2-cjohnson30*
 - b. Select this new *srv2-cjohnson30* folder.
 - c. Make sure you now see this change in the Location field. (Example: *Z:/OSM620/Virtual Machines/srv2-cjohnson30*)
 - iii. Click **Next**.
6. On the "Specify Disk Capacity" screen, do the following:
 - i. Maximum disk size (GB): **80**
 - ii. Select *Split virtual disk into multiple files*.
 - iii. Click Next.
7. On the "Ready to Create Virtual Machine" screen, do the following:
 - i. Click on **Customize hardware...**
8. On the new "Hardware" screen, do the following:
 - i. Select *Memory*, and change the value to: **4096**
 - ii. Select *Processors*, and change:
 - a. Number of processors: **2**
 - b. Number of cores per processor: **1**
 - c. Virtualize Intel VT-x/EPT or AMD-V/RVI: **Checked**
 - d. Virtualize CPU performance counters: **Unchecked**
 - e. Virtualize IOMMU (IO memory management unit): **Checked**
 - iii. Select *Network Adapter* and confirm:
 - a. *Connected at power on*: **Checked**
 - b. *NAT*: **Checked**
 - iv. Click **Close**.
9. Back in the "Ready to Create Virtual Machine" screen, click **Finish**.
10. The virtual machine should launch.
11. If you get a dialog box about *Side channel mitigations*, check the box for *Do not show this hint again* and click **OK**.
12. Click **Close**.
13. Click on the big **Play** to turn on the virtual machine begin the OS installation. This may take some time.

Part 2: Post-Installation Tasks

Investigation 7: VM3 Installation - Windows 11 Client

Name: client-cjohnson30 Name: client1-cjohnson30 RAM: 4GB CPU: 2 processors Storage: 64 GB

Part 1: Setup Instructions

In the main window, you should see a large + symbol icon titled **Create a New Virtual Machine**. Click it. 2. In the new dialog box, keep *Typical* selected and click the **Next** button. 3. On the next screen, *Guest Operating System Installation*,

do the following:

1. Select *Installer disc image file (ISO)*:
2. Now click **Browse**.
 - i. Navigate to where you saved your **Windows 11 Education** downloaded ISO and select it.
3. Once selected, the previous screen should now say "*Windows 11 x64 detected.*" If it doesn't, you haven't selected the right file, or your download was corrupted. Ask for help.
4. Click **Next**.
5. On the "Name the Virtual Machine" screen, do the following:
 - i. Virtual machine name: *client1-senecausername*
 - a. For example, if my Seneca e-mail address is *cjohnson30@myseneca.ca*, then my Seneca username is *cjohnson30*. This would give me a VM name of *client1-cjohnson30*.
 - ii. Location: If using an external SSD (like with our lab computers), click **Browse** and navigate to your external SSD.
 - a. Create the following directory structure in your SSD: OSM620 > Virtual Machines
 - b. Select this new Virtual Machines folder.
 - c. Make sure you now see this change in the Location field. (Example: Z:/OSM620/Virtual Machines)
 - iii. Click **Next**.
6. On the "Encryption Information" screen, do the following:
 - i. Select *Only the files needed to support TPM are encrypted*.
 - ii. Enter the same password as your other VMs in both fields.
 - iii. **Uncheck** the *Remember the password on this machine in Credentials Manager* option.
 - iv. Click **Next**.
7. On the "Specify Disk Capacity" screen, do the following:
 - i. Maximum disk size (GB): **64**
 - ii. Select *Split virtual disk into multiple files*.
 - iii. Click **Next**.
8. On the "Ready to Create Virtual Machine" screen, do the following:
 - i. Click on **Customize hardware...**
9. On the new "Hardware" screen, do the following:
 - i. Select *Memory*, and change the value to: **4096**
 - ii. Select *Processors*, and change:
 - a. Number of processors: **2**
 - b. Number of cores per processor: **1**
 - c. Virtualize Intel VT-x/EPT or AMD-V/RVI: **Checked**
 - d. Virtualize CPU performance counters: **Unchecked**
 - e. Virtualize IOMMU (IO memory management unit): **Checked**
 - iii. Select *Network Adapter* and confirm:
 - a. *Connected at power on*: **Checked**
 - b. *NAT*: **Checked**
 - iv. Click **Close**.
10. Back in the "Ready to Create Virtual Machine" screen, click **Finish**.
11. The virtual machine should launch.
12. If you get a dialog box about *Side channel mitigations*, check the box for *Do not show this hint again* and click **OK**.
13. Click **Close**.
14. Click on the big **Play** to turn on the virtual machine begin the OS installation. This may take some time.

Part 2: Installation Options

1. Once the Windows 11 Setup screen appears:
2. On *Select language settings*, keep the defaults and click **Next**.
3. On *Select keyboard settings*, keep the defaults and click **Next**.
4. On *Select setup option*, do the following:
 - i. Select *Install Windows 11*
 - ii. Check the box next to *I agree everything will be deleted, including files, apps, and settings*
 - iii. Click **Next**.
5. On the *Product key* page, enter your product key and click **Next**.
6. On *Applicable notices and license terms*, click **Accept**.
7. On *Select location to install Windows 11*, keep the defaults and click **Next**.
8. On *Ready to install*, click **Install**.
9. The Windows Installer will now install the OS. This may take some time, and the percentage may freeze at certain points. Be patient.
10. When the installer finishes, Windows 11 will start up and you will be launched into the First-Run Setup. Go to Part 3.

Part 3: First-Run Setup

1. On *Is this the right country or region?*, select **Canada** and click **Yes**.
2. On *Is this the right keyboard layout or input method?*, stick with the default and click **Yes**.
3. On *Want to add a secondary keyboard layout?*, click **Skip**.

Windows will now check for available updates from the Internet. If it finds any, it will install them automatically.

This process may take some time. Please be patient. Your VM may restart on its own.

Part 4: Account Creation

Now that language/keyboard and installer updates have been applied, it's time to create your account.

1. On *Let's set things up for work or school*, select **Sign-in options**.
2. On this next screen, click on **Domain join instead**.
3. On *Who's going to use this device?*, enter your **Seneca username**, (*Not* your full name), then click **Next**.
4. On *Create a super memorable password*, enter the same password you've used for your other VMs and click **Next**.
5. Confirm it on the next screen and continue.
6. On *Now add security questions*, fill out three security questions, clicking **Next** after each.
7. On *Let Microsoft and apps use your location*, select **No**, then click **Accept**.
8. On *Find my device*, select **No**, then click **Accept**.
9. On *Send diagnostic data to Microsoft*, scroll down to select **Required only**, then click **Accept**.
10. On *Improve inking & typing*, select **No**, then click **Accept**.
11. On *Get tailored experiences with diagnostic data*, select **No**, then click **Accept**.

Windows will now check for *more* available updates. As before, if it finds any, it will install them automatically.

This process may take some time. Please be patient. Your VM may restart on its own.

Once complete, you will be presented with a login screen. Move to the next part to continue.

Investigation 8: Post-Installation Tasks - Windows 11

In this investigation, we'll log in for the first time and run through several post-installation tasks both necessary and for user comfort.

Part 1: First Login

1. Enter your password to login. First-login may take a few minutes as your profile is set up.

Part 2: Installing VMware Tools

1. Once presented with the desktop, the very first thing we'll do is install the **VMware Tools** to integrate better with the application.
2. Outside of the VM window, way at the top of the actual VMware Workstation application, look for the **VM** menu item. (The order at the top is *File, Edit, View, VM, Tabs, Help*, if you're having trouble finding it.)
3. Click the following: **VM > Install VMware Tools**
4. Ignore the "AutoPlay" dialog box that pops up. It can take a while and disappears quickly.
5. Instead, open the **Start Menu**, and click on **File Explorer**.
6. In the new File Explorer window, look to the left-side menu bar.
7. Scroll down to *DVD Drive* and click it.
8. Inside *DVD Drive: VMware Tools*, find and double-click on the file: **setup64**
9. On the *User Account Control* dialog box that pops up, click **Yes**.
10. Wait for the installer to load.
11. On the first screen of the installer, click **Next**.
12. On *Choose Setup Type*, select **Complete**, then click **Next**.
13. On *Ready to install VMware Tools*, click **Install**.
14. This may take a few minutes, and the screen may flash a few times.
15. When complete, the last screen has a **Finish** button. Click it. (You have no choice!)
16. The installer will then ask you if you'd like to restart your computer. Click **Yes**.

Part 3: Setting the Time Zone

1. Login again.
2. Find the time on the bottom right of the screen (in the VM, not your host machine!)
3. Right-click on the time and select **Adjust date and time**.
4. Check the displayed time and time zone.
5. The time should match current time, and the time zone should say "(UTC-5:00) Eastern Time (US & Canada)".
6. If the time zone is wrong, change it to the value above.
7. Confirm your time zone and time matches local time.

Part 4: Setting Internal Computer Name

When we created our VM, we gave it the name *client1-SenecaUsername*. This only applies to how VMware Workstation

sees the VM, not to how the internal Windows OS sees itself. We need to change that to match.

1. In the *Settings* window you still have open (or reopen it from the Search bar), look to the left-hand menu bar and select **System**.
2. The very first thing you see at the top is the current computer name. This is what we're going to change.
3. Click on the **Rename** text link.
4. In the *Name your device* field, enter your VM's name: **client1-SenecaUsername** (replacing *SenecaUsername* with your actual username)
5. Click **Next**.
6. You will now be asked if you'd like to restart. Click **Restart now**.
7. After restarting and logging back in, go into *Settings > System* and confirm your computer name is now **client1-SenecaUsername**.

Part 4: Windows Updates

A critical part of a security-conscious mindset is running regular updates. **This is NOT something you do only once at the start of installation.** You should be running these regularly to keep up to date with security fixes and zero-day exploits.

1. If you're already in the *Settings* application, look through the left menubar for **Windows Update** and click it.
2. In the *Windows Update* main screen, scroll down to **Advanced Options** and click it.
3. The very first option is *Receive updates to other Microsoft products*. Toggle this from **Off** to **On**.
4. At the top of the screen, where it says *Windows Update > Advanced Options*, click **Windows Update** to go back to the previous screen.
5. You will likely already see updates ready. Click on **Download & install all**.
6. As you might expect, this can take a while. Timing will depend on your Internet connection, how fast your computer is, how fast your SSD is, and how many updates there are. Please be patient. Your computer may restart.
7. Once updates have begun, take a break while it does its thing. Grab a drink, make a sandwich, text a friend.
8. After updates are complete, go back into *Windows Update* and click **Check for updates** again. There may be (and often times are) more.
9. If there are more updates, complete Steps 5-8 again until there are no more updates available.
10. In *Windows Update*, scroll back down to **Advanced options** again and click it.
11. Inside *Windows Update > Advanced options* scroll down to **Optional updates** and click it.
12. Select all available updates that appear (you may have to expand some lists).
13. Click **Download & install**.
 - Windows Activation
 - Theme Changes (Dark Mode)
 - Internet Connectivity Check w/IE
 - Install Firefox and Make Default
 - Slim Down Firefox and Change Search
 - Remove Copilot

Lab 2 - Security and Remote Connectivity

Lab Preparation

Purpose / Objectives of Lab 2

In this lab, you will conduct several Windows system administration tasks to secure your servers against would-be attackers and gain preliminary experience with the command line interface.

If you encounter technical issues, please contact your professor via e-mail or in your section's Microsoft Teams group.

Minimum Requirements

Before beginning, you must have:

1. Successfully completed Lab 1.
2. Attended the Week 3 lecture
3. Read through the Week 3 slides, and have them handy as a reference for concepts.
4. Your external SSD (or personal computer) with your VMs from Lab 1.
5. Your VM login credentials.
6. Optional, but recommended: Caffeine delivery system.

Key Concepts

Security: From the Beginning

In the not-too-distant past, companies would focus on getting their product and systems working and relegating security as their last step, often as an afterthought. When security is only considered at the end of a project, it's very difficult to remember all the ways in which your product interacts and things can get missed.

This created several high-profile breaches in the 90s and early 2000s, and our approach to security had to be reconsidered.

As a result, we now consider security **from the beginning**. As you create applications, add users to databases, create links between services, you *must* keep security in mind at every step of development. Securing as you go is the best method, but even something as simple as simply documenting unsecured parts of your code as you go can be enough (assuming you go back and fix them!)

Generally, we apply the concept of **Principle of Least Privilege** to security. Essentially, this boils down to locking everything down as much as possible and only allowing what and who you need through. Open access makes you a target. You'll be applying this principle to the firewall later in this lab.

We also take a look at defaults. Most systems and software come with pre-configured defaults to make out-of-the-box setup easy. This can take the form of a default username and password, default ports, etc. In a well secured system, these are often changed to avoid hack attempts. If you know the default, there's a high chance that hackers know it as well. You'll be changing some defaults in this lab.

This is not an exhaustive list of applied security, but it does give use a bit of working knowledge. You'll need it for this lab as well as in our later work.

Firewalls

In short, **a firewall is a utility that sits on your computer between your network connection and the rest of your system.**

Any application, service, or other data that is sent or received by your computer goes through your firewall first. The dominant network protocol is TCP/IP, which means we're dealing with *packets*.

A firewall looks at these packets.

To be clear, the firewall doesn't look *inside* packets, but just at the outside data like IP address and/or port destination, etc. The actual transmitted data is still secure and unread.

Generally with firewalls, we apply the *Principle of Least Privilege* by dropping all new connections by default, and allowing a few exceptions. This is known as **whitelisting**.

Editing Text Files

As you will sometimes be working in the Windows PowerShell command line environment, it is useful to learn a least one common method of editing text files.

Although programmers and developers usually use graphical IDE's to code and compile programs (Visual Studio, Sublime, Eclipse, etc), they can create source code using a text editor and compile their code directly on the server to generate executable programs (without having to transfer them for compilation or execution).

Developers very often use a text editor to modify configuration files. In this course, you will become familiar with the process of installing, configuring, and running network services. Text editors are an important tool to help this setup, but are also used to "tweak" or make periodic changes in service configurations.

The most readily-available command line text editor built into Windows is **Notepad**.

However, Notepad is not available in Server Core. To edit a text file in that environment, we have three main options:

1. Use PowerShell's object-oriented programming to send an edit directly into a text file. This is cumbersome and is not interactive (you don't see the text file on screen), but it is the only built-in option.
2. Transferring the file to a different computer that does have a text editor (like your Windows 11 client), modifying the file there, and transferring it back to your Server Core machine.
3. Using **Visual Studio Code** to connect remotely the the environment. This is by far the coolest and most convenient option.

We'll be doing all three in this lab to show you how, but going forward, VS Code will be our go-to for interacting with Server Core in most instances. (You will be tested on all three options, so don't skip them!)

Investigation 1: Windows Defender (Firewall)

In this investigation, we're going to take a look at the Windows Defender firewall to see how firewall rules can be applied to secure our servers and let the few things we want to allow through.

It should be noted that the Windows Defender firewall *should not be your only*

defense. Production environments will often have managed networks that restrict access on a larger scale, some of which you will get into with your networking courses.

These work together, though we will only be focusing (mostly) on the Windows firewall in this course.

Part 1: Windows Server GUI (srv1) - Configuring the IIS Role

On this server, we will configure our first server role, IIS (a web server). This will be used only for testing connectivity between your other VMs and working with the firewalls.

We will spend more time with Server Roles in a later lab. For now, simply follow the instructions.

1. In the *Server Manager* application, in the menu bar on the left side of the window, click on **Local Server**.
2. Now, at the top right of the window, click **Manage > Add Roles and Features**.
3. The *Add Roles and Features Wizard* dialogue box pops up.
4. On the first page, *Before you begin*, check the box next to **Skip this page by default**, then click **Next**. This allows us to skip this page any other time we want to add a role or feature in later labs.
5. On the *Installation Type* page, select **Role-based or feature-based installation** and click **Next**.
6. On the *Server Selection* page, select **Select a server from the server pool**, then **srv1-username**, and finally **Next**.
 - i. Note: In later labs, we'll be able to target other servers for remotely installing features. Here, we're selecting the local server.
7. On the *Server Roles* page, this is where we'll select our IIS web server. Scroll

through the listing until you find **Web Server (IIS)**, check the box next to it, and click **Next**.

8. A secondary dialogue box, *Add features that are required for Web Server (IIS)?*, pops up. Make sure the **Include management tools (if applicable)** box is checked, then click **Add Features**.
9. Back in the *Server Roles* page, click **Next** again.
10. On the *Features* page, leave all settings as their defaults and click **Next**.
11. On the *Web Server Role (IIS)* page, click **Next**.
12. On the *Role Services* page, keep all defaults and click **Next**.
13. Finally, on the *Confirmation* page, click **Install**.
14. This may take a few minutes. Please be patient.
15. When the wizard has the status message "*Installation succeeded on srv1.*", setup is complete and you can click **Close**.
16. Verify your local web server works by opening *Firefox* and going to the following address: **127.0.0.1**
17. If you see the **Internet Information Services** splash page, you've successfully completed the installation and can move on to the next part!

(Insert instructions for loading this page on the Windows 11 client to test connectivity.) (Insert instructions for loading this page on Windows Server Core if possible, even just unrendered HTML code.)

Part 2: Testing VM Network Connectivity

As part of your Lab 1 environment setup, you tested your connection to the Internet from each of your 3 virtual machines.

It's now time to test if we can use each VM to connect to the others.

1. On your Windows Server GUI (srv1) VM, open Command Prompt.
2. Grab this VM's IP address with the following command and write it down:

`ipconfig`

3. On your Windows Client (client 1) VM, open Command Prompt.
4. Enter the following command to test our ability to talk to srv1: `ping "srv1-ipaddress"` where *srv1-ipaddress* is the IP address from Step 2.
 - i. Example: `ping 192.168.1.14`
5. It doesn't work, does it? That's normal at this stage and we'll fix it below. For now, we'll try connecting to our web server on srv1. That should show us the two VMs can talk to each other.
6. Open *Firefox* on your **Windows 11 Client (client1)** and use the IP address from Step 2 (your srv1 IP address) as the URL.
 - i. Example: `http://192.168.1.14`
7. Can you see the IIS splash page that you saw in Part 1, Step 17? If so, you have connectivity!
8. Continue to Part 3.

Part 3: Windows Server GUI (srv1) - Applying Firewall Rules

We will now apply our security-conscious policy by configuring our firewall on this server.

At the moment, the server's firewall is configured using defaults. As mentioned above, defaults are a security risk as they are known to everyone and can be used against you. If an attacker *doesn't* know your configuration, it's harder for them to know what's open and what's not and how to attack.

As you saw from Part 2, ping between client1 and srv1 didn't work. This is because, by default, the ability to ping a server is turned off. A ping (a type of ICMP packet) is typically used to see if you can get a response from a server. Having a server respond lets an attacker know there's a machine there that they can then try to break into.

We want to turn on ping so we can test connections between our machines, but we have to be careful. Turning that on through the firewall too broadly opens us up to that vulnerability.

We are going to turn it on *only* for our local network. Our VMs will be allowed to ping each other, but anything outside of our subnet (192.168.1.0/24) can't. Best of both worlds.

To do this, we're going to work with the **Windows Defender Firewall** on srv1.

1. In the *Server Manager* application, go to *Local Server*.
2. In the *Properties* section of this page, look for the **Microsoft Defender Firewall** line item.
3. Next to the line, it should say **Public: On**. Click on this.
4. The *Windows Security > Firewall & Network protection* application opens.
5. On this screen, you will see three networks: *Domain network*, *Private network*, and *Public network*.
6. We'll spend more time with these in later weeks, but for now, all three should say: **Firewall is on**.
7. Below this, click on the link that says: **Advanced settings**
8. This opens the *Windows Defender Firewall with Advanced Security* application.
9. On the first page, you'll notice the same three profiles: *Domain Profile*, *Private Profile*, and *Public Profile*.
10. All three have the same overall rules:
 - i. **Windows Defender Firewall is on**: This just confirms the firewall is active for this network profile.
 - ii. **Inbound connections that do not match a rule are blocked**: This means that all incoming connections and requests (like trying to ping this machine) are blocked *by default*. If you want to allow a certain type of connection or service into your server, you have to make a specific

rule for it. This is whitelisting and our *Principle of Least Privilege* in action.

- iii. **Outbound connections that do not match a rule are allowed:** This means all network data leaving the server is automatically allowed. The logic here is that if the server is the one deciding the send out information, it's likely fine. (Note: The only time we ever whitelist outbound connections is for specialized security settings like government compliance.)


11. At the moment, we're dealing with the *Public Profile* context. Let's allow ICMP ping!
12. On the left-hand menu bar, click **Inbound Rules**.
13. This loads a ton of already-defined rules. Thankfully, the one we need has already been defined. We just need to turn it on.
14. At the top of this area, click **Name** to order the list by name (this is not the default).
15. Now, scroll down until you can find the following and double-click it: **Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In), Private, Public**.
 - i. **Note:** There are two of these rules! Look for the one that has **Private, Public** under the Profile header. The wrong one says Domain.
16. This opens this rule's configuration window, starting on the *General* tab.
17. In the *General* tab, look for the following and check the box next to it:
Enabled
18. Click **Apply**, then **OK**.
19. Back in the main *Inbound Rules* window, you should see that same rule line now says **Yes** under the *Enabled* header.
20. You've enabled ICMP ping! Let's go test it.
21. Switch back to your client1 machine, and run the same ping you ran earlier pointing to srv1. (Refer to *Part 2, Step 4*.)
22. Does it work?

23. If it does, congratulations! You've just enabled ping for connectivity checking to *srv1* and gone through your first foray into the Windows Firewall.

i A note for later labs: By default, this rule is set to only allow incoming pings (ICMP requests) from computers on your **VM network**—that is, other virtual machines on the same VMware NAT or host-only network as your server—not from the wider Internet or any physical computers in the classroom.

You can see this by checking the **Scope** tab in the rule's properties, where "Remote IP address" is set to "Local subnet."

We'll spend more time on how **Scope** and **Private/Public profiles** affect your firewall rules in our *Secondary Network* assignment.

 **Lab Question:** Why did we *not* have to do this for our IIS web server setup?

In the **Inbound Rules** list, scroll through to see if you can find the rule that's allowing web server pages to be requested from *srv1*.

Write down the name of the rule in your Lab Logbook when you find it (it's not called IIS) and explain why you think you didn't have to enable this rule yourself. Think back to when you installed the IIS Server Role.

Part 4: Windows Server Core (srv2) - Applying Firewall Rules

Let's apply the same incoming ping firewall rule to our Server Core machine so we can check its network connectivity as well.

1. Login to your Server Core (*srv2*) machine.
2. The *sfconfig* text-based application automatically launches.
3. Select option 8 to find this machine's IP address. Write it down.

4. Back in your Windows 11 Client (client1), try to ping this address. Does it work?
5. Just as in srv1, it doesn't.
6. Go back to srv2.
7. If you're still in the *Network settings* page, leave the field blank and hit **Enter** to go back to the main screen.
8. Select option 15 to exit to PowerShell.
9. As there's no GUI, we need to use PowerShell for firewall management.
10. Let's take a look at the existing rules, just like in Part 2. Run the following PowerShell command:

```
Get-NetFirewallRule | Where-Object DisplayName -Like '*ICMPv4-In*'
```

11. Several results appear in the search. Look for the one with the name: **Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)**
12. This firewall rule object, starting from *Name* all the way down to *PackageFamilyName* lists all the configured properties for this rule.
13. Read through these properties. Recognize any from our GUI version?
14. This rule matches both our incoming ping requests and keeps to *Private*, *Public* profiles. As with our GUI version, the scope is defaulted to local subnet only.
15. Let's turn on this incoming firewall rule by running the following command:

```
Enable-NetFirewallRule -Name CoreNet-Diag-ICMP4-EchoRequest-In
```

16. This selects the right rule and enables it.
17. One of the things you **must** get into the habit of doing with CLI commands is ***double-checking your work***.
18. Let's do that now by asking the system if it was actually enabled:

```
Get-NetFirewallRule -Name CoreNet-Diag-ICMP4-EchoRequest-In
```

19. Look for the *Enabled:* field. See how it's changed to **True**? It worked!
20. Optional: We can even check the scope for local subnet only as we did in the GUI version if we want with the following command:

```
Get-NetFirewallRule -Name CoreNet-Diag-ICMP4-EchoRequest-In | Get-NetFirewallAddressFilter
```

Output:

```
LocalAddress   : Any
RemoteAddress  : LocalSubnet4
```

21. Last, because there are two incoming ICMP rules (*Domain* profile and *Private, Public* profile), let's check that only the *Private, Public* rule is enabled:

Command:

```
Get-NetFirewallRule -DisplayName "Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)" | Select-Object DisplayName, Profile, Enabled
```

22. This shows a brief status of both rules. The *Domain* version should show **False** under the *Enabled* header, while the *Private, Public* version should show **True**.

Output:

```
DisplayName
Profile Enabled
-----
-----
Core Networking Diagnostics - ICMP Echo Request
(ICMPv4-In)           Domain    False
Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)
Private, Public      True
```

23. Now, let's test our ping. Switch over to your Windows 11 Client (*client1*).
24. Open a Command Prompt window, and run the following command: `ping`
`srv2-ipaddress` (Refer to Part 3, Step 3).
25. Does it work?
26. If it does, congratulations! You've just enabled ping for connectivity checking to *srv2* and gone through your first foray into the PowerShell!

Part 5: Windows Client (*client1*) - Applying Firewall Rules

Finally, let's enable ping on our Windows Client machine.

1. Login to *client1*.
2. Click the **Start** button and type the following search: **firewall**
3. Of the options that appear, select: **Windows Defender Firewall with Advanced Security**
4. Just as in *Part 2* with *srv1*, navigate to **Inbound Rules**.
5. Click the *Name* header to sort by name.
6. Find the following rule:
 - i. Name: **Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)**
 - ii. Profile: **Private, Public**

7. Double-click it to open the rule's configuration settings.
8. In the *General* tab, find and check the box next to **Enabled**.
9. Click **Apply**, then **OK**.
10. Open a *Command Prompt* window and run the following to get your client's IP address: `ipconfig`
11. Switch to your *Server Core (srv2)* machine.
12. In PowerShell, run the following command: `ping client1-ipaddress` (Where client1-ipaddress is the address from Step 10.)
 - i. Example: `ping 192.168.1.15`
13. Does it work?
14. If it does, congratulations! You've just enabled ping for connectivity checking to *client1* and have full connectivity checking for your entire environment! This will become **very** handy in Labs 3-4.

Investigation 2: Remote Management - Windows Server GUI (srv1)

While you have direct access to all three VMs from VMware, these days most machines are remote. This means you typically do not have direct, physical access to the servers.

Even if you do, having remote access to allow you to manage your servers from the comfort of your office (or home!) instead of walking to each physical machine is far better.

In this investigation, we'll set up remote access to our Windows Server GUI (srv1) so we can connect to it using your Windows 11 Client VM.

Part 1: Enabling Remote Desktop Connections

In this part, we'll turn on Remote Desktop (RDP) on the server. This will allow you to remotely connect to *srv1* from your Windows 11 Client.

1. In the *Server Manager* application, navigate to *Local Server*.
2. In the main *Properties* window, find the line entry for: **Remote Desktop**
3. Click the link next to it: **Disabled**
4. In the new *System Properties* window, you should automatically be on the *Remote* tab.
5. Look down to the section on **Remote Desktop**.
6. Toggle the option for: **Allow remote connections to this computer**
7. Ensure the checkbox next to this is on: **Allow connections only from computers running Remote Desktop with Network Level Authentication (recommended)**
8. Click **Apply**, then **OK**.

Part 2: Adding your RDP Firewall Rules

In this part, we'll add a firewall rule to allow the connection over the local network.

1. On *srv1*, open **Windows Defender Firewall with Advanced Security**.
2. In the *Inbound Rules* section, look for the following rules:
 - i. Remote Desktop - Shadow (TCP-In)
 - ii. Remote Desktop - User Mode (TCP-In)
 - iii. Remote Desktop - User Mode (UDP-In)
3. If they're all enabled (they should be), you're good to go!
4. Why check? Always check your firewall rules when enabling a new service, even if it *usually* enables the associated firewall rule automatically. **Never**

assume!

5. If these rules are **not** enabled, enable them now.

Part 3: Connecting to Windows Server (srv1) from Windows 11

In this part, we'll verify our work by connecting to the server using our Windows 11 client VM.

1. Make sure you've got your *srv1* IP address from earlier. (You wrote it down, *right?*)
2. On *client1*, click the **Start** menu.
3. Search for and open: **Remote Desktop Connection**
4. In the newly opened application, in the *Computer* field, enter the IP address for *srv1*.
5. Click **Connect**.
6. Enter your *srv1* username and password, then click **OK**.
7. A security dialogue box pops up. **Check the box next to "Don't ask me again for connections to this computer" before the next step.**
8. Click **Yes**.
9. If you can see your *srv1* desktop, congratulations! You now have remote access.
10. To quit the remote session, find the floating HUD at the top of the screen with the IP address and click on the **X** icon.
11. Note: **This will not shutdown your *srv1* VM. It only ends your remote session.**
12. Note 2: A remote session will automatically lock your VM's direct session. You will need to unlock it again when you go back to it through VMware Workstation directly.

Investigation 3: Remote Management - Windows Server Core (srv2) with Remote Desktop

In this investigation, we'll set up remote access to our Windows Server Core (srv2) so we can connect to it using your Windows 11 Client VM.

Part 1: Enabling Remote Desktop Connections

In this part, we'll turn on Remote Desktop (RDP) on the Core server (srv2).

1. Login to *srv2*.
2. If you're at the PowerShell command prompt, run the following command to get back into the text-based *Server Manager* application: `sconfig`
3. Select option 7.
4. In the *Remote desktop* page, select option **E** to enable Remote Desktop.
5. It will ask you to select a security level. Select **Option 1**.
6. Press **Enter** to complete and go back to the main *Server Manager* page.
7. **Double-check your work.** Select option 7 again.
8. Is the *Remote desktop status* changed and correct?
9. If not, go through Steps 4-9 again.
10. If yes, press **Enter** without entering any options to go back to the main screen without making any changes.

Part 2: Adding a RDP Firewall Rules

In this part, we'll check that the RDP firewall rules have been enabled.

1. Select **Option 15** to exit to PowerShell.
2. Run the following PowerShell command to check the status of your RDP firewall rules:

Command:

```
Get-NetFirewallRule | Where-Object { $_.DisplayName -like '*Remote Desktop*' -and $_.Direction -eq 'Inbound' } |  
    Select-Object Name, DisplayName, Profile, Enabled
```

Output:

Name	DisplayName
Profile Enabled	
----	-----
RemoteDesktop-In-TCP-WS	Remote Desktop - (TCP-WS-
In) Any False	
RemoteDesktop-In-TCP-WSS	Remote Desktop - (TCP-WSS-
In) Any False	
RemoteDesktop-Shadow-In-TCP	Remote Desktop - Shadow (TCP-
In) Any True	
RemoteDesktop-UserMode-In-TCP	Remote Desktop - User Mode (TCP-
In) Any True	
RemoteDesktop-UserMode-In-UDP	Remote Desktop - User Mode (UDP-
In) Any True	

3. Check that your output matches the above. This is the same as on *srv1*, just at the command line. Notice which are enabled and which are not?
4. If you need to, grab the IP address for *srv2* here by running this command in PowerShell: `ipconfig`

Part 3: Connecting to Windows Server (srv2)

from Windows 11 via RDP

In this part, we'll verify our work by connecting to *srv2* using our Windows 11 client VM. This is essentially the same as our instructions for *srv1*, but with a different IP address.

1. Make sure you've got your *srv2* IP address from earlier. (You wrote it down, *right?*)
2. On *client1*, click the **Start** menu.
3. Search for and open: **Remote Desktop Connection**
4. In the newly opened application, in the *Computer* field, enter the IP address for *srv1*.
5. Click **Connect**.
6. Enter your *srv2* username and password, then click **OK**.
7. A security dialogue box pops up. **Check the box next to "*Don't ask me again for connections to this computer*" before the next step.**
8. Click **Yes**.
9. If you can see your *srv2* Command Prompt window, congratulations! You now have remote access.
10. To quit the remote session, find the floating HUD at the top of the screen with the IP address and click on the **X** icon.
11. Note: **This will not shutdown your *srv2* VM. It only ends your remote session.**
12. Note 2: A remote session will automatically lock your VM's direct session. You will need to unlock it again when you go back to it through VMware Workstation directly.

Part 4: File system navigation

(A primer on file system navigation in Windows. Have them create a few

directories and empty files using CLI only.)

Part 5: Editing a file

(This is where we do the single example of using PowerShell to modify a text file.)

Investigation 4: Remote Management - Windows Server Core (srv2) with SSH

Part 1: Enabling SSH Connections

In this part, we'll turn on incoming SSH connections so we can connect to our server using Visual Studio Code from the Windows 11 client.

1. Login to your Server Core (*srv2*).
2. Exit the *Server Manager* program by selecting **Option 15**.
3. In PowerShell, run the following command to install the OpenSSH Server

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

4. After it installs, we have to start the SSH service. Run the following:

```
Start-Service sshd
```

5. We now have to set the SSH service to start automatically every time the system boots. Run the following:

```
Set-Service -Name sshd -StartupType 'Automatic'
```

```
# Install OpenSSH Server
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

# Start and enable the service
Start-Service sshd
Set-Service -Name sshd -StartupType 'Automatic'

# Confirm firewall rule is present or add it
Get-NetFirewallRule -Name *ssh*
# If not present, add:
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)'
-Enabled True -Direction Inbound -Protocol TCP -Action Allow
-LocalPort 22
```

Part 2: Adding an SSH Firewall Rule

1. We'll now add an SSH rule. One already exists, but it's only attached to the *Private* profile and would not work for us.
2. Instead, we'll create a custom rule. Run the following command:

```
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)'
-Enabled True -Direction Inbound -Protocol TCP -Action Allow
-LocalPort 22 -Profile Private,Public
```

3. This rule sets the following:
 - i. Name: sshd
 - ii. Display Name: OpenSSH Server (sshd)
 - iii. Enabled: True
 - iv. Profile: Private, Public
 - v. Direction: Inbound

- vi. Action: Allow
 - vii. Protocol: TCP
 - viii. Port: 22
4. Verify your work! First, let's look at the firewall object by running the following command:
 5. Now, let's verify the protocol and port with the following command:

Command:

```
Get-NetFirewallRule -Name 'sshd'
```

Output:

```
Name : sshd
DisplayName : OpenSSH Server (sshd)
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Private, Public
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully
from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local
RemoteDynamicKeywordAddresses : {}
```

Command:

```
Get-NetFirewallRule -Name 'sshd' | Get-NetFirewallPortFilter |  
Select-Object Name, Protocol, LocalPort
```

Output:

Name	Protocol	LocalPort
-----	-----	-----
	TCP	22

6. If your own output looks like the above, congratulations! If not, try your configuration commands again or ask for help.

Part 3: Connecting to Windows Server Core (srv2) from Windows 11 via SSH

1. On your Windows 11 Client (*client1*), open Command Prompt.
2. Enter the following command: `ssh Administrator@srv1ipaddress` (Where *srv2ipaddress* is your actual *srv2* IP address.)
 - i. Example: `ssh Administrator@192.168.1.15`
3. The first time you connect, a prompt will ask you if you are sure. Type **yes** and hit **Enter**.
4. Type your password when asked and hit **Enter**.
 - i. Note: You will *not* see asterisks or other characters as you type. **This is normal**. It is taking your keyboard input. It may take you a few tries to get used to it.
5. Once logged in, you are now on *srv2*'s Command Prompt environment.
6. To get back into PowerShell (which you should), run: `powershell`
7. It is **not** recommended to run the *sconfig* program over an SSH connection.

If you need to use the *sconfig* / *Server Manager* text-based program, connect through your RDP connection in Part 1.

i A note about SSH and *Server Manager*:

Your SSH connection will allow you to run the *sconfig* Server Manager program and load it. However, many of its functions fail over an SSH connection because it relies on background interactive processes that can't run on an SSH session. They must be run from the RDP connection, despite it *looking* the same.

For an example of this, run `sconfig` from your SSH session and try to run Windows Updates. If it finds any and you try to install them, it will fail. This is an SSH+sconfig problem; run the same thing through RDP and it will work fine.

Investigation 5: Remote Management - Windows Server Core (srv2) with Visual Studio Code + SSH

Part 1 : Connecting to srv2 with Visual Studio Code

(insert instructions on how to use the UI to connect, including saving the connection for future use)

Part 2: A Quick Tour of VS Code + Remote SSH

(A brief tour of the three panes: File navigation, text editor, and PowerShell terminal)

Part 3: VS Code for Fun and Profit

(Brief exercises having them do some file management with the navigation pane, opening and modifying a file, and then running a PowerShell command)

(Insert caveat about most **sconfig** commands not working through SSH or VS Code, they must be done through RDP because of how interaction works.)

Assignment 1

Assignment 2