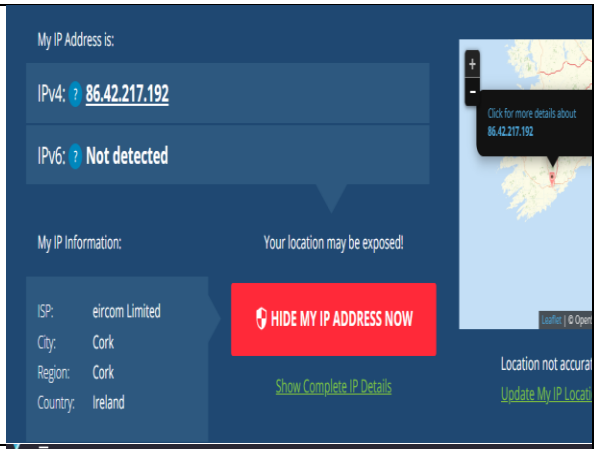
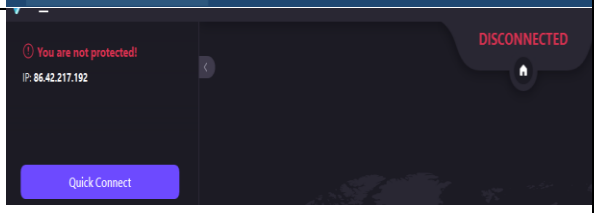
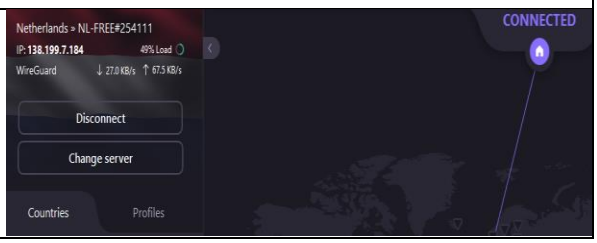
# Importance of Encryption & Secure Communication

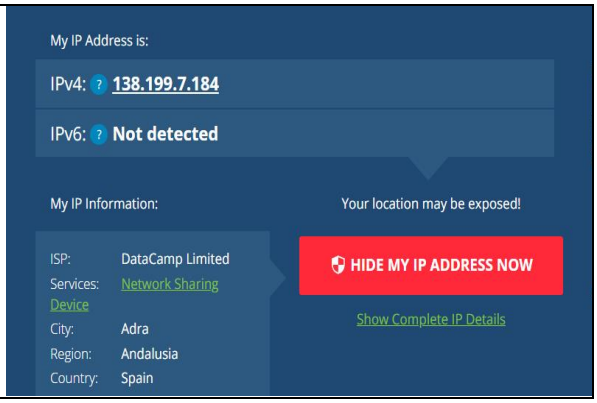
Daniel Sheehan

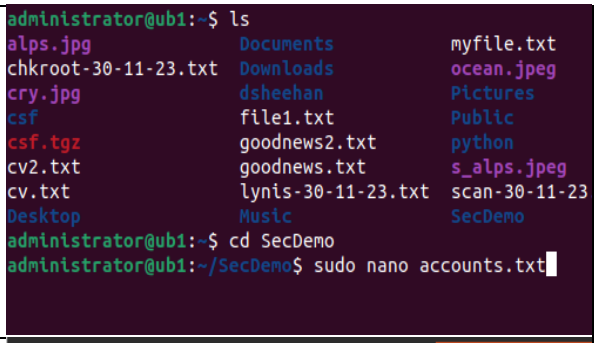
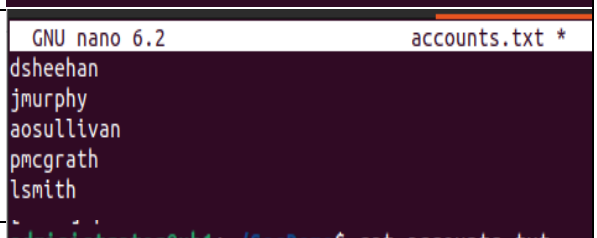
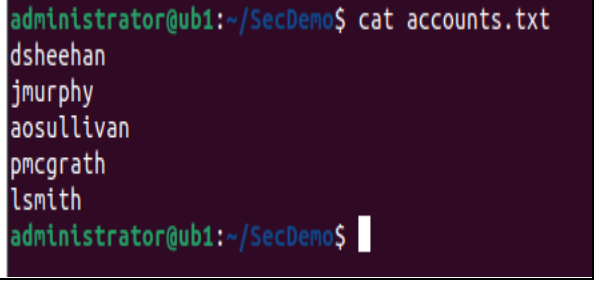
**Introduction:**

In this skills demo I will be exploring the importance of secure digital communication as well as a variety of encryption methods such as using a VPN to safely browse the web, Encryption methods such as GPG and Steghide, setting up and configuring secure wireless communication through encryption methods such as WPA2 and MAC address filtering. Another key aspect of secure communication I will be covering is the ability to use SSH to connect to a remote server and using SFTP to securely transfer files onto ymy local machine.

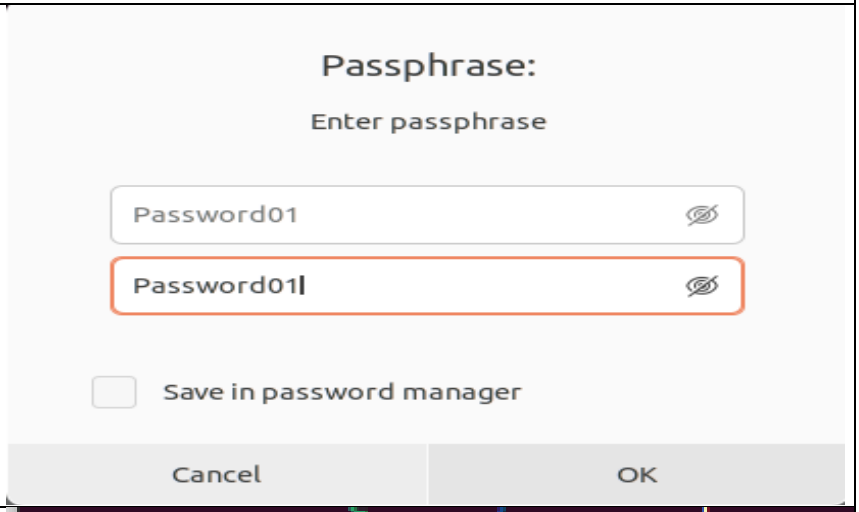
## Virtual Private Networks (ProtonVPN)

| | |
|---|---|
| The first thing I will do for this opening segment of the skills demo is check my IP address using https://whatismyipaddress.com.<br><br>As I can see my IPV4 address is 86.42.217.192 and is coming out of Cork, Ireland |  |
| I will be using the free edition ProtonVPN in order to obtain a new IP address that will be assigned to us from another country.<br><br>Once the software is installed press quick connect on the map screen. |  |
| I should now see that the indicator in the top right has changed to purple from red and now says “Connected” meaning I have successfully been assigned a new IP address. |  |

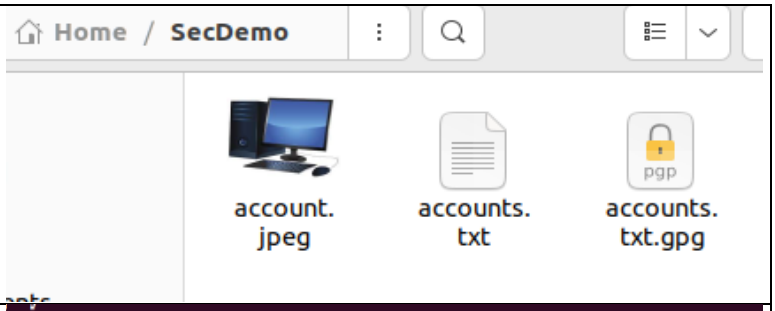
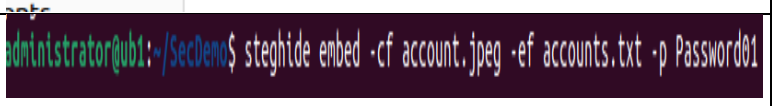
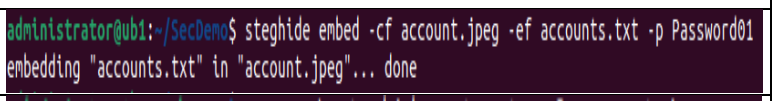
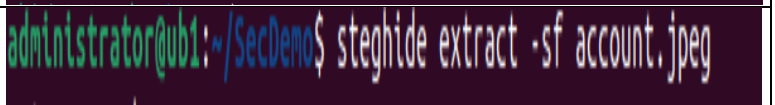
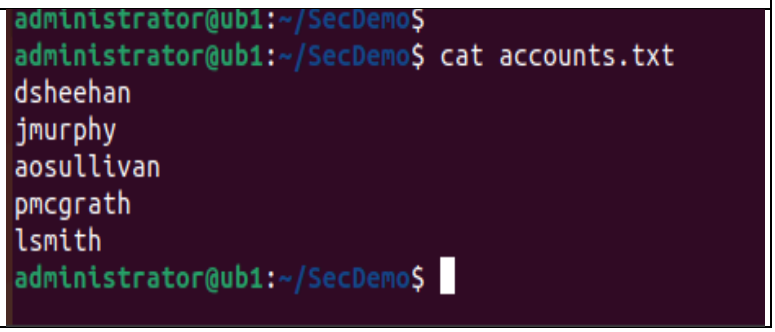
| | |
|---|---|
| Upon navigating back to https://whatismyipaddress.com and refreshing the page I can now see that I have been assigned a new IP address (138.199.7.184) coming from Andalusia, Spain<br><br>This shows us that my VPN is working and I are now ready to browse the internet more securely. |  |

| | |
|---|---|
| I will create a text file in my SecDemo directory known as accounts.txt<br><br>Sudo nano accounts.txt |  |
| In this NANO file I will enter a few sample account names for the contents |  |
| I will then cat my accounts.txt file to verify the contents |  |

# Encrypt (GPG)

| | |
|---|---|
| I will now encrypt my accounts.txt file:<br>gpg –symmetric accounts.txt | `administrator@ub1:~/SecDemo$ gpg --symmetric accounts.txt` |
| I will now set the password as:<br><br>Password01<br><br>This will ensure my file stays protected. |  |
| I now have 2 files the original and the encrypted (accounts.txt.gpg) | `accounts.txt  accounts.txt.gpg`<br>`administrator@ub1:~/SecDemo$` |
| I can now see from the cat output the file contents are encrypted. | `administrator@ub1:~/SecDemo$ cat accounts.txt.gpg`<br>`?'ϭY◆o◆◆V◆◆nro◆ra◆z◆&il3◆Ku!◆7Z◆◆_F◆#◆◆◆◆`<br>`7◆◆◆H◆◆◆◆W◆]O◆◆◆◆◆◆;◆◆◆◆◆◆◆administr` |
| Remove the original file | `rm accounts.txt` |
| keep the encrypted version so I can revert it back to accounts.txt | `administrator@ub1:~/SecDemo$ ls`<br>`accounts.txt.gpg` |
| I have now decrypted the contents of my encrypted file and outputted them to a new file called accounts.txt | `administrator@ub1:~/SecDemo$ gpg -d -o accounts.txt accounts.txt.gpg`<br>`gpg: AES256.CFB encrypted data`<br>`gpg: encrypted with 1 passphrase` |
| As I can see from the output I have successfully decrypted the text content and put it into my accounts.txt file | `administrator@ub1:~/SecDemo$ gpg -d -o accounts.txt accounts.txt.gpg`<br>`gpg: AES256.CFB encrypted data`<br>`gpg: encrypted with 1 passphrase`<br>`administrator@ub1:~/SecDemo$ ls`<br>`accounts.txt  accounts.txt.gpg`<br>`administrator@ub1:~/SecDemo$ cat accounts.txt`<br>`dsheehan`<br>`jmurphy`<br>`aosullivan`<br>`pmcgrath`<br>`lsmith` |

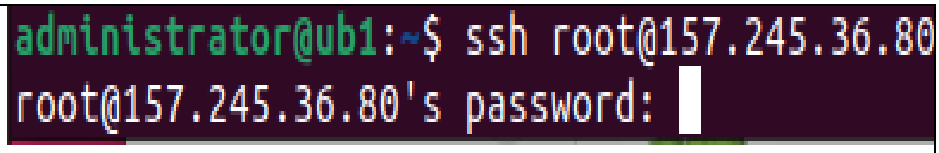
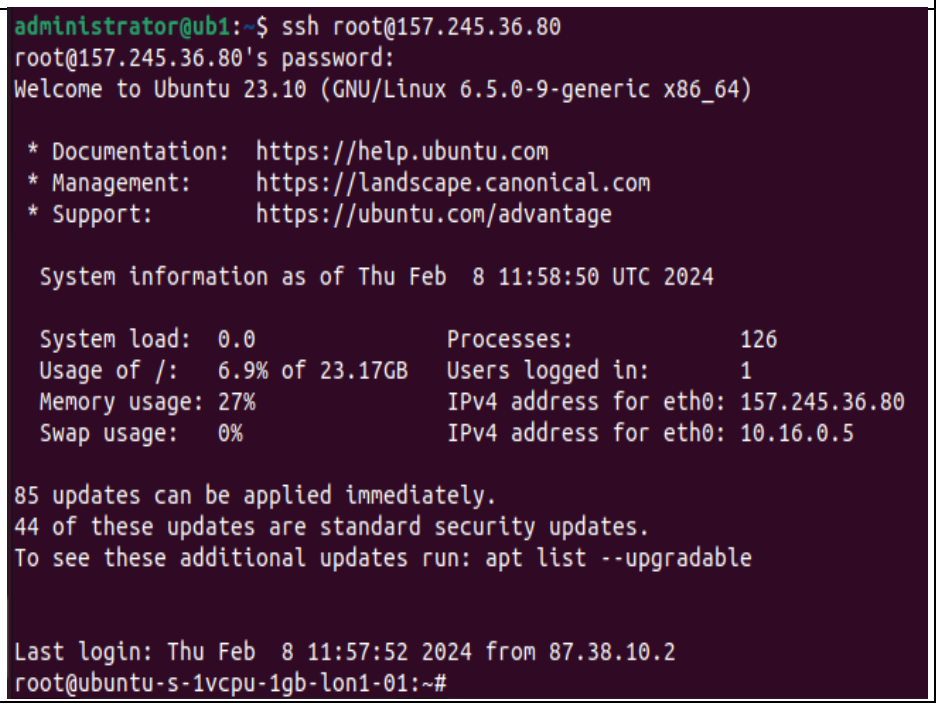
# Steghide (embed text into image)

| | |
|---|---|
| I will now download a jpg image to hide my plain text file (accounts.txt) this means I are embedding my text in the image. |  |
| To do this I will be using steghide<br><br>Steghide embed -cf account.jpg -ef accounts.txt -p Password01 |  |
| My text is now embedded in my jpeg file. |  |
| Let's decrypt this.<br><br>Using Password01 I will extract my txt file from my jpeg. |  |
| As shown here I have successfully made a new accounts.txt file that should contain my decrypted text content |  |
| I will now cat this to verify the output.<br><br>As shown by the output I can see the text has been decrypted telling us I have successfully completed these steps. |  |

# Connecting to a remote server

In this next segment I will be connecting to a remote digitalocean server in a data centre in London through SSH, once I have connected, I will be creating a folder that will contain a txt file once I have created both files, I will be using the list (ls) command to show the contents of the directory. Once this has been completed, I will be using SFTP to transfer the file from the virtual server machine to my local machine and deleting the folder on the server.

**SSH:**

| | |
|---|---|
| To begin I will navigate to the cli terminal on my Ubuntu machine and type the following commands in order to connect to this remote server being hosted on a data centre in London.<br><br>Ssh root@157.245.36.80<br><br>Once I type this command I will then enter the password 4BlindMice and attempt to connect |  |
| As I can see I have successfully connected to the remote server as the root user meaning I have permissions to create new directories and files.<br><br>I will type cd to ensure I are in the home directory |  |

| | |
|---|---|
| Now that I know where I are I will make a new directory called “daniel” by using the mkdir command.<br><br>After I create the directory I will type ls again and I should see the daniel directory has been created. | root@ubuntu-s-1vcpu-1gb-lon1-01:~# mkdir daniel<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# ls<br>Abdi Ali Rawan damien daniel ewa jb mateusz<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# |
| Next I will cd into the daniel directory and create a file called cv.txt using the command sudo nano cv.txt | root@ubuntu-s-1vcpu-1gb-lon1-01:~# ls<br>Abdi Ali Rawan damien daniel ewa jb mateusz<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# cd daniel<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~/daniel# sudo nano cv.tx |
| I will write contents to verify the integrity of my file and then press ctrl s ctrl x to save and exit | GNU nano 7.2 cv.txt *<br>This is daniels CV |
| Once I type ls again I can see my cv.txt file | root@ubuntu-s-1vcpu-1gb-lon1-01:~# cd danie<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~/daniel# l<br>cv.txt |
| I will now press exit to close my connection to the server. | root@ubuntu-s-1vcpu-1gb-lon1-01:~/daniel# exit<br>logout<br>Connection to 157.245.36.80 closed.<br>administrator@ub1:~$ |

**SFTP(Secure File Transfer Protocol):**

| | |
|---|---|
| I will now log back into the server using SFTP and the same login as previosuly:<br><br>root@157.245.36.80<br><br>pass: 4BlindMice | administrator@ub1:~$ sftp root@157.245.36.80<br>root@157.245.36.80's password:<br>Connected to 157.245.36.80.<br>sftp> |

| | |
|---|---|
| Once I have connected I will us the ls command to view my servers home directory containing the daniel directory. | Connected to 157.245.36.80.<br>sftp> ls<br>daniel jb<br>sftp> |
| Using the lls command I can see the contents of my local machine away from the server. I will use sftp to retrieve my cv.txt file from my server and put it on my local machine. | sftp> lls<br>404.gif music.zip<br>'books(1).csv' network-dump.flag.pcap<br>books.csv shark1.pcapng<br>cdDatabase-20230929 smile.png<br>cdDatabase-20230929.zip tree.jpeg<br>cds.csv 'Vocational Study Brief Daniel Sheehan.pdf'<br>giphy.gif 'Vocational Study Submission Daniel Sheehan.pdf<br>music<br>sftp> |
| To achieve this I must use the following command:<br><br>Get cv.txt<br><br>After this I will run another lls and as I can see my file has been retried and is now present on my local machine meaning I have used sftp successfully. | sftp> get cv.txt<br>Fetching /root/cv.txt to cv.txt<br>sftp> lls<br>404.gif music<br>'books(1).csv' music.zip<br>books.csv network-dump.flag.pcap<br>cdDatabase-20230929 shark1.pcapng<br>cdDatabase-20230929.zip smile.png<br>cds.csv tree.jpeg<br>cv.txt 'Vocational Study Brief Daniel Sheehan.pdf'<br>giphy.gif 'Vocational Study Submission Daniel Sheehan.pdf'<br>sftp> |
| I will now clean up after myselves in order to securely remove any traces of my activity. | sftp> cd<br>sftp> ls<br>cv.txt daniel jb<br>sftp> |
| Exit sftp | sftp> exit<br>administrator@ub1:~$ |
| I will login again using ssh and type ls where I can see my daniel directory. | root@ubuntu-s-1vcpu-1gb-lon1-01:~# cd<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# ls<br>Abdi cv.txt daniel jb |

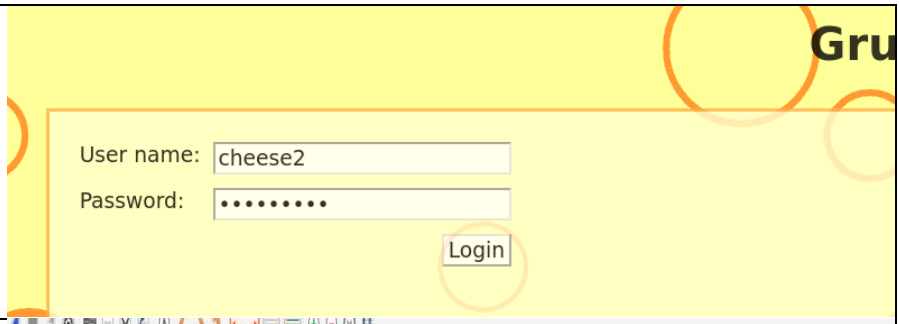
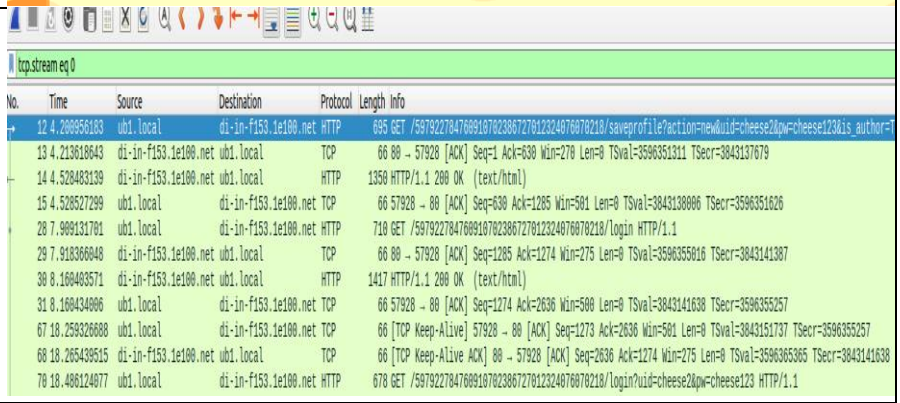
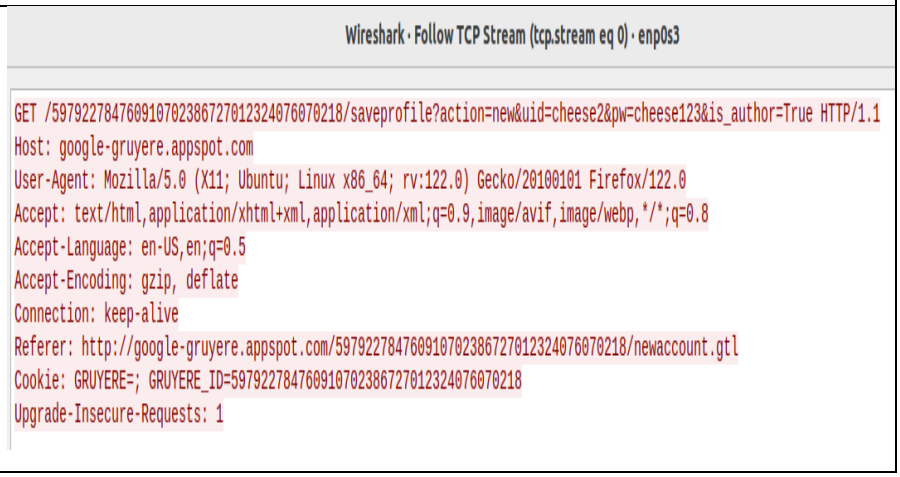
| | |
|---|---|
| Finally I will type rm -rf daniel<br><br>I will then type ls and see I have successfully removed the daniel directory meaning I have cleared any trace of my activity. | root@ubuntu-s-1vcpu-1gb-lon1-01:~# ls<br>Abdi cv.txt daniel jb<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# rm -rf daniel<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# ls<br>Abdi cv.txt jb<br>root@ubuntu-s-1vcpu-1gb-lon1-01:~# |

## HTTP vs HTTPS

In this segment of my skills demo I will be demonstrating the difference of a HTTP vs a HTTPS Ibsite. HTTP messages are written in plaintext which allows anyone on the internet to access and read them, HTTPS hoIver transmits all data encrypted, this means that when users fill out forms and submit sensitive information, they can be confident that it is secure.

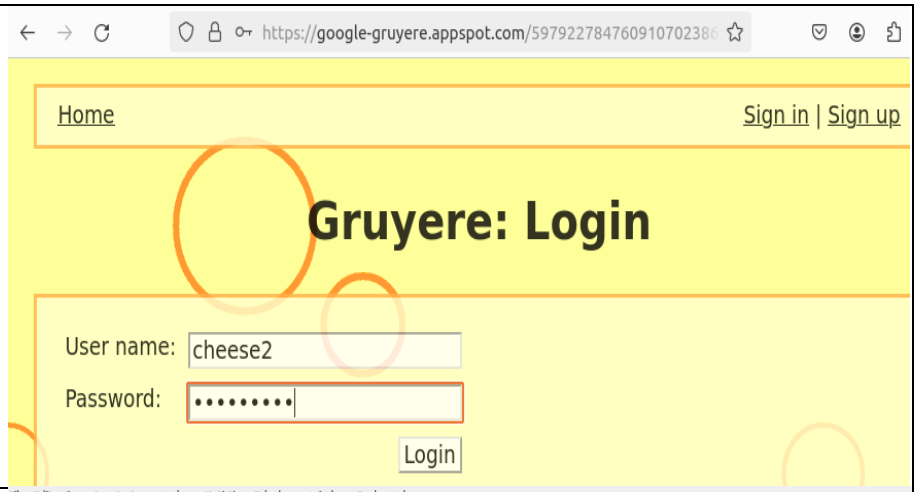
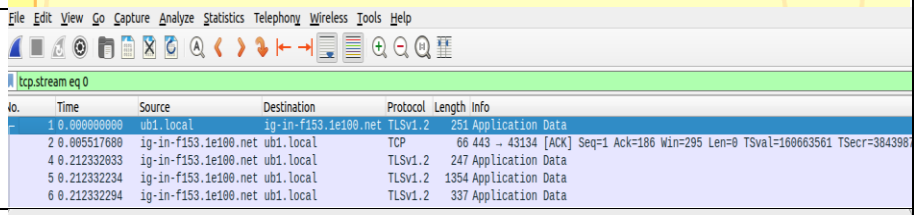
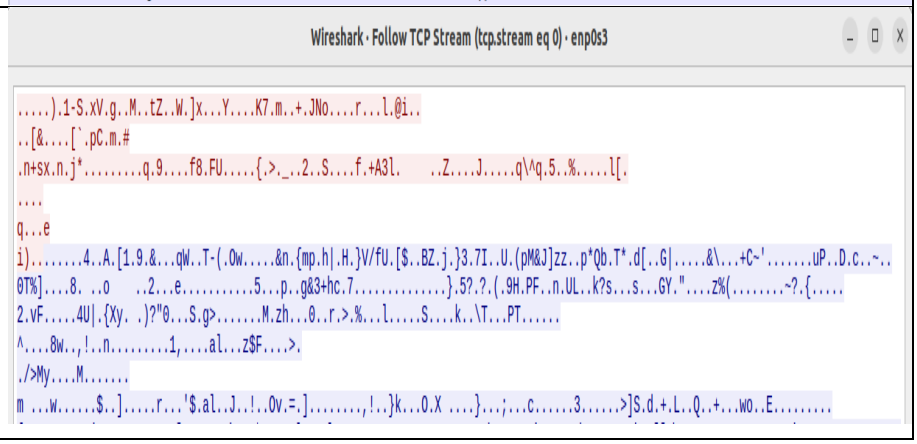
HTTP:

| | |
|---|---|
| I began by logging onto Google-Gruyere.appspot.com, I will use this site to look at the difference betIen http and https. |  |
| On the HTTP site I can see that there is a red mark going through the padlock this is warning users that the site may be unsecure and to be mindful when entering any data and filling forms |  |

| | |
|---|---|
| I will test this by creating and logging onto an account as follows:<br>User – cheese2<br>Password – cheese123 |  |
| Using Wireshark, I will monitor any packets going through the site and locating the one containing my username and password. Once I find it I will follow the TCP stream on the HTTP smyce and view the results |  |
| Once I follow the stream, I can clearly see my username (cheese2) and password (cheese123) can be vield as plain text by anyone monitoring the page. This shows us that no encryption method has been used as previously stated. |  |

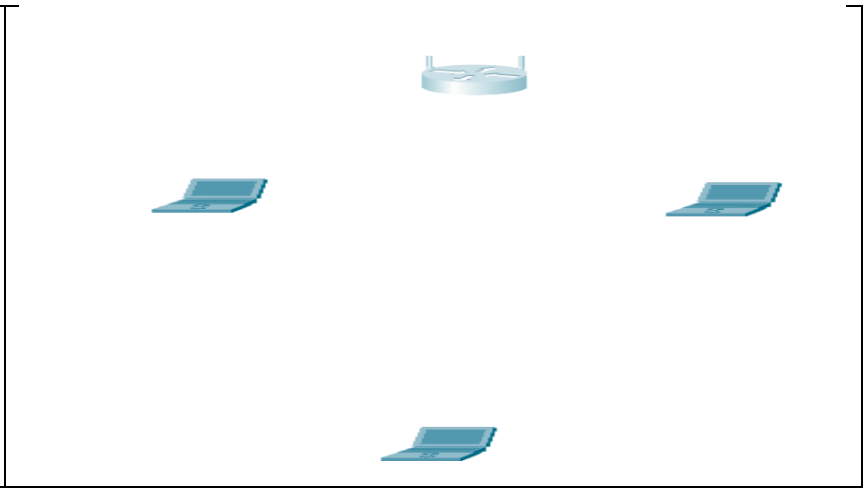
HTTPS:

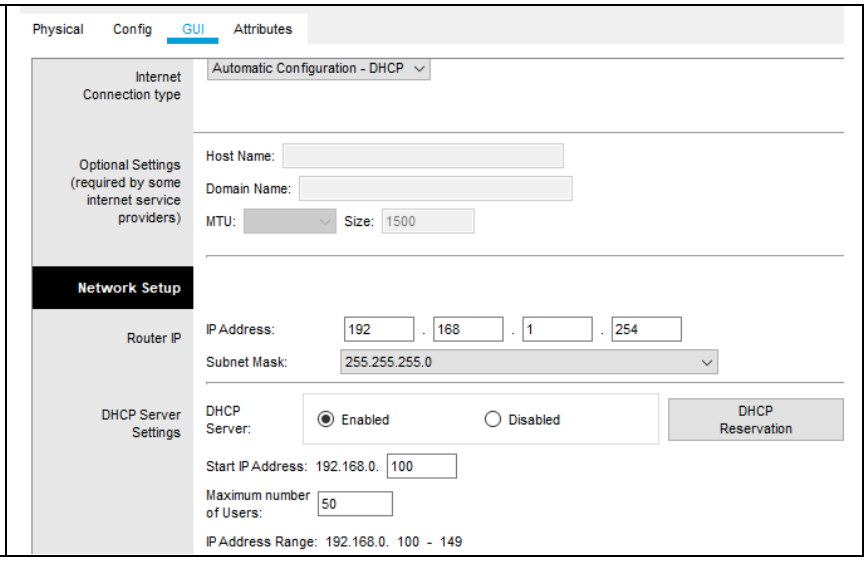
| | |
|---|---|
| Next, I will be testing the HTTPS version of the Gruyere site in order to see if my user inputted data is encrypted. |  |
| As I can see the padlock has no red line across it indicating that this version of the site is secure and encrypted. |  |

| | |
|---|---|
| I will repeat my previous steps and log in to the site using the following details:<br><br>User – cheese2<br><br>Password-cheese123 |  |
| Once I have entered my data, I will follow the TCP stream in order to see if my data is encrypted. |  |
| Here I can now see that my username and password are no longer displayed as plain text, this is because HTTPS has been used and my data is encrypted and more secure instead of being stored as plain text. |  |

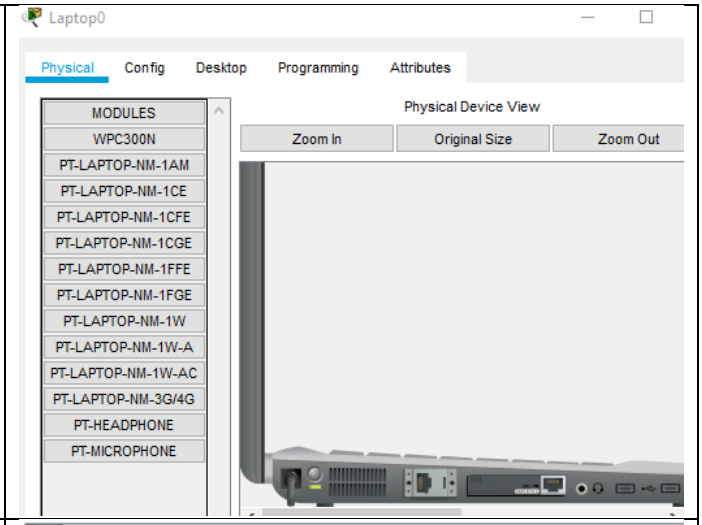
## Filtering MAC Address + WPA2 integration

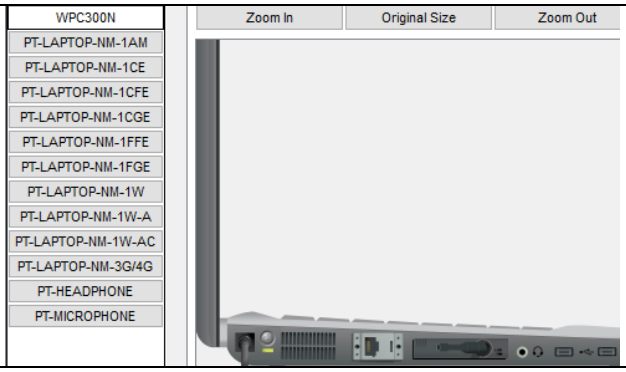
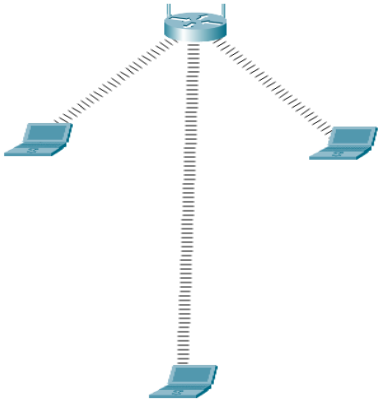
Topology & Wireless Router Setup:

| | |
|---|---|
| For this segment of the skills demo I will be implementing WPA2 on a wireless access point and utilizing MAC address filtering.<br><br>I will be setting up a topology in Packet Tracer using 3 laptops and Wireless Router |  |

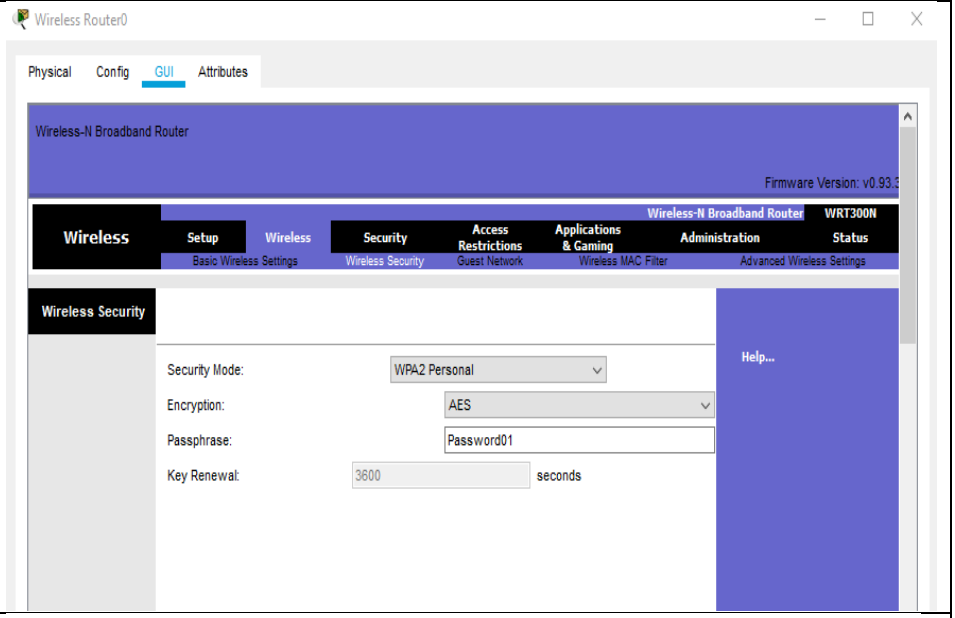
| | |
|---|---|
| I will begin by navigating to the Router's GUI and setting its IP address to 192.168.1.254 and ensuring that DHCP is enabled. My starting IP Address will also be set to 100 and I will have a maximum of 50 users. Once this is completed, I are ready to move onto the next step. |  |

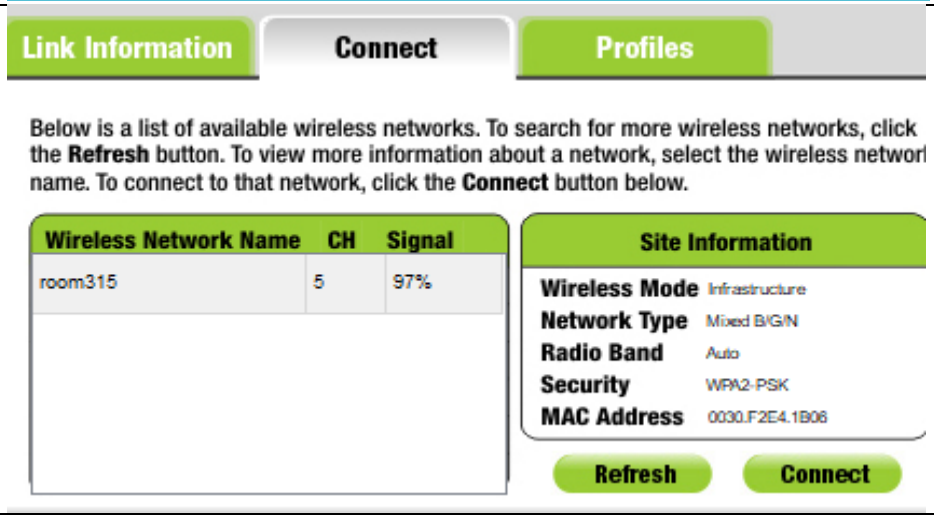
Laptops setup:

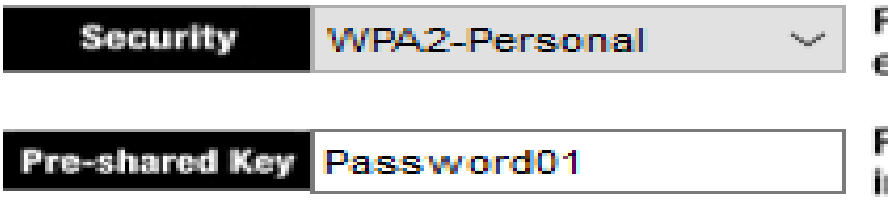
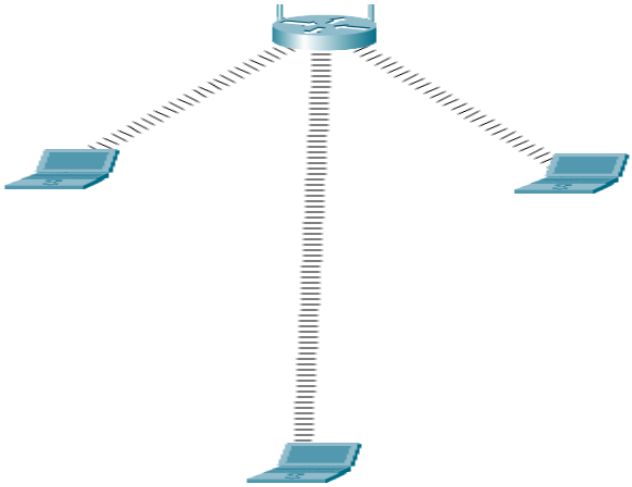
| | |
|---|---|
| Next, I will navigate to my 1st laptop and access the physical tab, here I will be met with a list containing a few different Wi-Fi card models. |  |
| Before selecting the card that I require I will have to click the button on the left of the laptop in order to poIr off the device. I will then remove the current card before I progress. |  |

| | |
|---|---|
| Next, I will insert the WPC300N card from the list and polr the laptop back on, and repeat these exact steps for all laptops/devices |  |
| I now have a successful connection for each device to the wireless router as indicated by the line which forms connecting each device to the wireless router.<br><br>Now that I can see my devices have connected successfully, I will need to set up a basic layer of encryption so that only authorized devices can connect to the network. |  |

Basic Encryption for Network (WPA2):

| | |
|---|---|
| I will now use Wi-Fi protected access (WPA2) in order to protect my network.<br><br>To do this I will navigate back to the GUI of my router and enable WPA2 Personal as my Security Mode. The password I will set is Password01. |  |
| I will have to scroll down and click save settings before exiting in order to save my changes. |  |

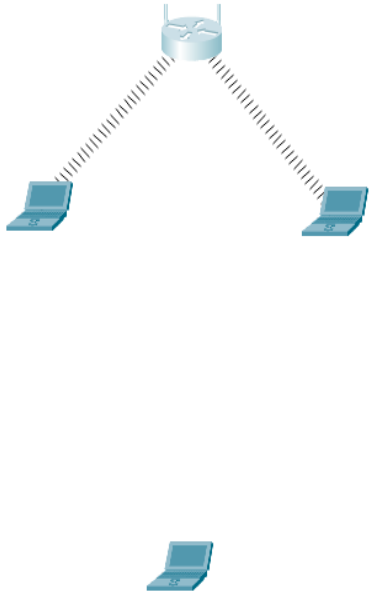
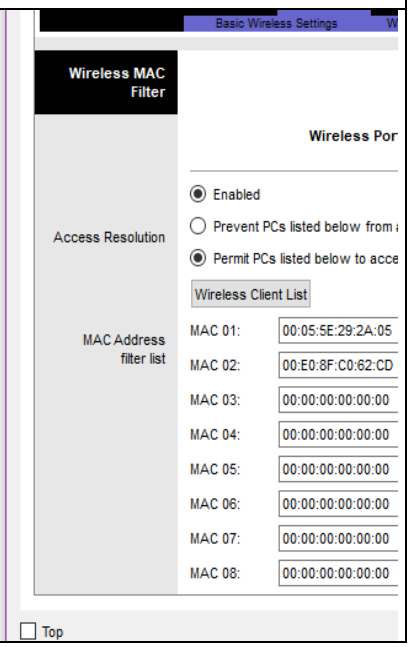
| | |
|---|---|
| My connection is gone so I must log in using the laptop GUI to access the router again otherwise I are not permitted to access the network.<br><br>This has shown us I have successfully enabled WPA2 |  |
| Going back to my Router GUI I will navigate to the basic wireless settings tab and configure the network as follows: |  |
| Next on the laptops I will navigate to the PC wireless tab using the GUI and get ready to join my Network that should now be set up. |  |
| Navigate to the second tab (Connect) and click refresh, Here I should see the name of my Wireless Network "room315" appear, Select it and say connect |  |

| | |
|---|---|
| Here I will need to enter the password I set up earlier: Password01 |  |
| Repeat for all devices and I have a connection again with a layer of security.<br><br>This shows that WPA2 has been utilized successfully. |  |

Filtering MAC addresses:

My next and final layer of security for my network that I will be setting up for this Skills Demo is MAC address filtering. This makes it so that only devices that have authorized MAC addresses from my Router have permission to connect to the network.

| | |
|---|---|
| In router GUI I will navigate to Wireless tab and select wireless MAC filter | Wireless \| Security \| Access Restrictions \| Applications & Gaming<br>ss Settings \| Wireless Security \| Guest Network \| Wireless MAC Filter |
| Here I have a list of mac addresses permitted to connect to the router, this offers an extra layer of security | Enabled / Disabled<br>Prevent PCs listed below from accessing the wireless network<br>Permit PCs listed below to access wireless network<br>Wireless Client List<br>MAC 01: 00:00:00:00:00:00 MAC 26: 00:00:00:00:00:00<br>MAC 02: 00:00:00:00:00:00 MAC 27: 00:00:00:00:00:00<br>MAC 03: 00:00:00:00:00:00 MAC 28: 00:00:00:00:00:00 |
| If I navigate to one of my laptops and enter CLI I can type ipconfig /all<br><br>This provides us with the devices MAC Address | C:\>ipconfig /all<br><br>Bluetooth Connection:(default port)<br><br>Connection-specific DNS Suffix..:<br>Physical Address.....: 0009.7C30.25EC<br>Link-local IPv6 Address.....: ::<br>IP Address.....: 0.0.0.0<br>Subnet Mask.....: 0.0.0.0<br>Default Gateway.....: 0.0.0.0<br>DNS Servers.....: 0.0.0.0<br>DHCP Servers.....: 0.0.0.0<br>DHCPv6 Client DUID.....: 00-01-00-01-A5-25-4D-93-00-05-5E-29-2A-05 |
| My MAC address: PC1 | Connection-specific DNS Suffix..:<br>Physical Address.....: 0005.5E29.2A05 |
| PC2 MAC Address: | Physical Address.....: 00E0.8FC0.62CD |
| PC 3 (Unpermitted MAC Address) | Physical Address.....: 0030.A300.36AA |
| For my next step I will navigate to the Router and under the wireless Tab I will click the "Wireless MAC Filtering" option this will provide us with a list that I can add the Physical MAC Addresses I have obtained from running the ipconfig command in CLI so that they are | Wireless \| Security \| Access Restrictions \| Applications & Gaming<br>s Settings \| Wireless Security \| Guest Network \| Wireless MAC Filter<br>MAC 01: 00:06:2A:40:9D:C8<br>MAC 02: 00:40:0B:98:C0:BA<br>MAC 03: 00:00:00:00:00:00 |

| | |
|---|---|
| permitted to join my network, I will purposely leave the address of the 3rd laptop out of the list in order to test if it has worked effectively | |
| As I can see I now have a list of MAC addresses that I have alloId to access the router, PC 1 and PC2 have access but PC3 can't connect as it's MAC address isn't listed among the permitted addresses |    |

This shows us that my list is working, and I have successfully filtered MAC addresses to allow and deny access to my Network depending on the devices MAC address and whether or not it is permitted to connect to my network.

**Conclusion:**

In conclusion in this skills demo I showcased a variety of different uses for secure communication such as setting up and using VPNs to browse the internet securely and setting up different encryption methods such as GPG and Steghide in order to keep data secure and hidden. I also managed to successfully set up secure wireless connections on 3 different devices using WPA2 and MAC address filtering to keep unauthorized devices off of my wireless network. I also learned the importance of remote server access through methods such as using SSH to access files and SFTP to securely transfer them from a network onto my physical way in an encrypted way. I also demonstrated the importance of HTTPS over HTTP and how users should be cautious when inputting data on unsecure sites in order to avoid theft and their data falling into the wrong hands.