

NSSA221 Systems Administration I Lab 06: Apache, Email, Logging, and Cron

INTRODUCTION

This lab is the coup de grâce of all labs! So far, you have learned how to manage users, devices, file systems, storage and version control. In this lab, you will explore services that are found in many enterprise environments as well as smaller organizations, web and email. We will also look at approaches to remote logging and managing system tasks using the Cron utility.

LAB SUMMARY

In this lab, you will use Windows 10 as the mail server, and Rocky Linux for a web server. The lab has three parts; the first part will have you configure an Apache web server. Once the server is running, you will create virtual hosts and produce a self-signed certificate to encrypt HTTP communications. Part 2 will explore the protocols and roles of various devices involved in email communication. The final part will give you experience running cron jobs and centralized logging in your environment.

GOALS

At the end of this lab, you will...

- Have a better understanding of the role and function of a web server in an enterprise environment.
- Have experience creating virtual hosts on a web server.
- Have experience creating self-signed certificates and using Transport Layer Security in HTTP communications.
- Have experience installing and configuring an email server.
- Acquire knowledge about email communication and the protocols in play.
- Have a better understanding of email communications.
- Have experience deploying and configuring remote logging.
- Have experience setting up the Cron service.

PREPARATION

- Read chapters 13 and 17 from the Linux Bible.
- Read chapters 7, 8, and 9 from Windows Server 2016 Inside Out.
- Read Cron Job: a Comprehensive Guide for Beginners 2019.

ACTIVITY SUMMARY

Activity 1 – Configuring an Apache Web Server

Activity 2 – Creating Virtual Web Servers

Activity 3 – Creating a Self-Signed Certificate

Activity 4 – Configuring Apache for Transport Layer Security

Activity 5 – Installing and Configuring MailEnable

Activity 6 – Email Client Configuration

[Activity 7](#) – Capturing Email Network Traffic

[Activity 8](#) – Remote Logging

[Activity 9](#) – Setting up the Cron Service

ACTIVITIES

Activity 1 – Deploying a Basic Web Server

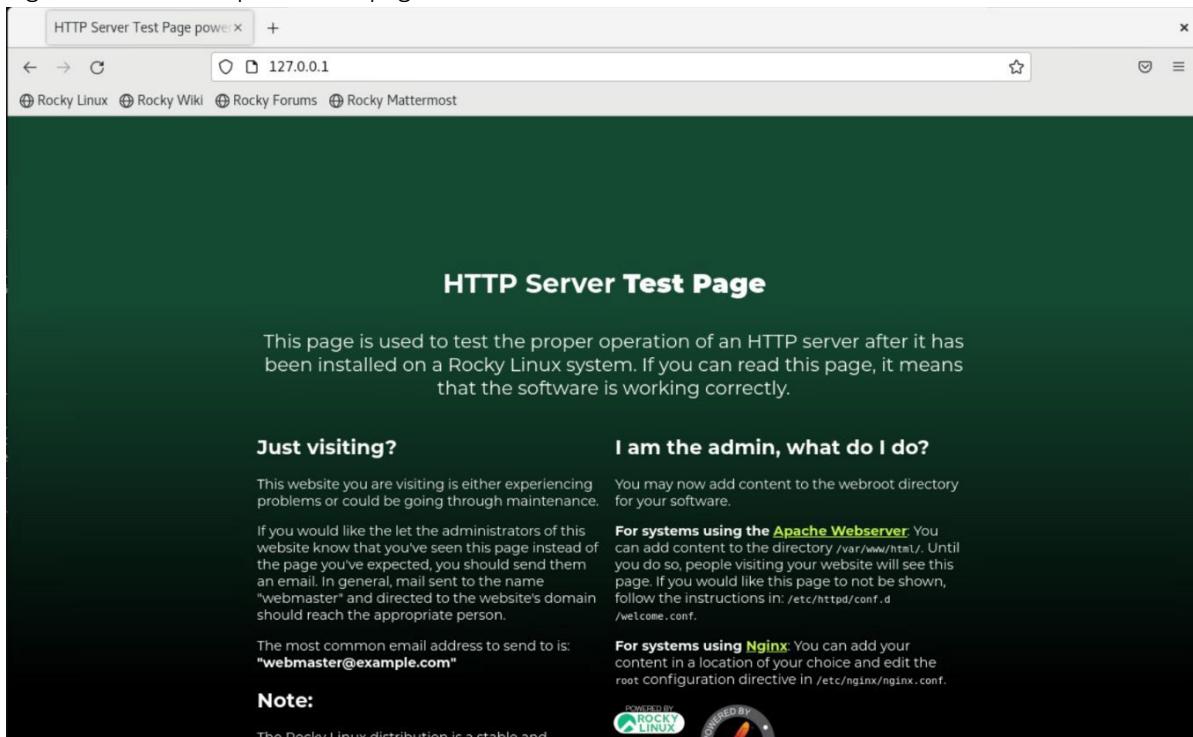
For this activity, you will be installing and configuring the Apache web service on the Rocky Linux 8 virtual machine in your environment. Feel free to “scale out,” another virtual machine or use an existing Rocky VM in your deployment.

- a. The Rocky Linux 8 should be assigned the same hostname from Lab 1 and have a DNS record in the DNS server. If you “scaled out,” a new deployment, make sure to update DNS and assign a hostname to the new virtual machine.
- b. In the previous lab, you had exercises involving SELinux, be mindful of the SELinux status of the deployment. You can edit the configuration file and set SELinux to permissive, or use the `setenforce` command to set it to permissive during runtime.
- c. Install the required packages using the following command.

```
$ dnf -y install httpd
```

- d. Start the httpd service (refer to the `systemctl` cheat sheet for help) and verify the installation by starting the `httpd` service and entering “`localhost`” into a web browser to bring up the default Apache webpage (Figure 1).
- e. Enable the HTTP service so that it starts when the system boots.

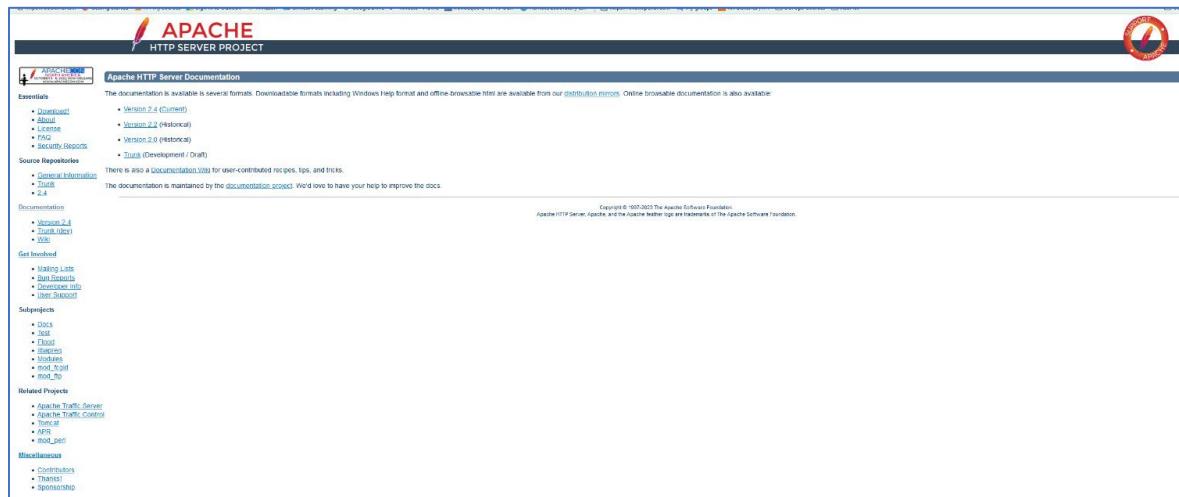
Figure 1 – Default Apache Webpage



- e. To view the Apache documentation, use the following URL <https://httpd.apache.org/docs/> (Figure 2).

RIT | Golisano College of Computing and Information Sciences School of Information

Figure 2 – The Official Apache Documentation

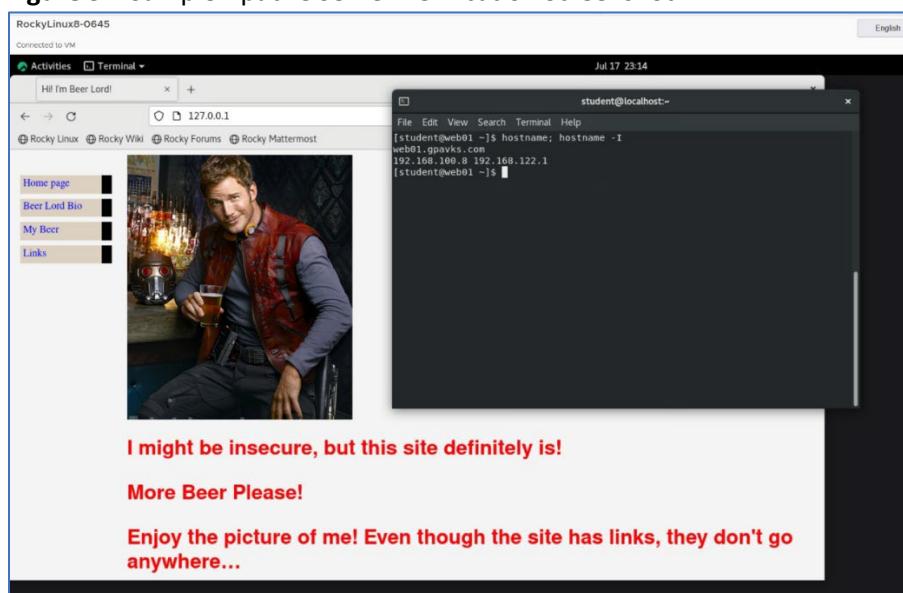


- f. Add an *index.html* file, to the default Apache directory, located in */var/www/html*. You may use the sample index file provided in myCourses, or create your own.
- g. Remember to restart the httpd service (**systemctl restart httpd**) after any changes are made to the configuration files and verify that the webpage loads correctly.



For the report, include a single screenshot showing your FQDN (the output of the **hostname** command), and the output of the **hostname -I** command. In the same screenshot, open a browser to launch the webpage. The figure must be a single screen shot properly labelled and included in the report. Refer to Figure 3 for an example.

Figure 3 – Sample Apache Server Verification Screenshot



Activity 2 – Creating Virtual Web Servers

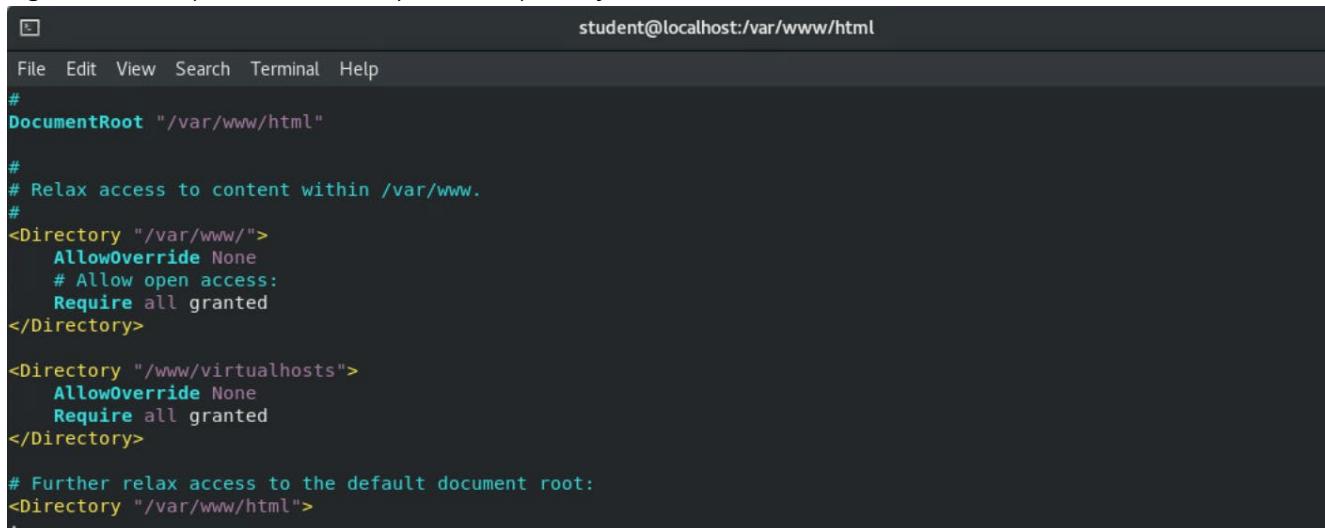
In an enterprise environment, it is unlikely that you will have multiple servers, physical or virtual, for each website. Instead, you will find a dedicated Apache server hosting many sites through virtual web servers. This activity is designed to introduce you to this concept by having you create three sites on a single (and in your case, a virtual) Apache webserver. One of the sites will be the default site, and the other two will be virtual host sites. The examples in these instructions use the following sites.

- www.gpavks.com
- starlord.gpavks.com
- gamora.gpavks.com

- a. Edit the *httpd.conf* file by adding a directive to allow access to the virtual hosts. In the example provided, the virtual hosts are in the */www/virtualhosts/* directory; you may choose a different directory; and if you do, make sure to check the permissions. We will worry about creating the directories later, for now; edit the *httpd.conf* file by adding the following directive. Figure 4 provides an example of an entry.

```
<Directory "/www/virtualhosts">
    AllowOverride None
    Require all granted
</Directory>
```

Figure 4 – Example Directive Entry in the *httpd.conf* File.



A screenshot of a terminal window titled "student@localhost:/var/www/html". The window shows the contents of the */etc/httpd/conf.d* directory. The files listed are *httpd.conf*, *startlord.conf*, and *starlord.conf*. The *httpd.conf* file contains the directive shown in Figure 4, and the *starlord.conf* file contains a similar directive for the *starlord.gpavks.com* site.

```
student@localhost:/var/www/html
File Edit View Search Terminal Help
#
# DocumentRoot "/var/www/html"
#
# Relax access to content within /var/www.
#
<Directory "/var/www/">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

<Directory "/www/virtualhosts">
    AllowOverride None
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
:
```

- b. Create the necessary configuration files for each of the virtual hosts in the */etc/httpd/conf.d* directory. To keep track of things, I recommend you name the file so that it associates to the site. For example, for "startlord.gpavks.com," I have named it "*starlord.gpavks.com.conf*". Yes, all I am doing is using the URL and adding ".conf" at the end. Trust me, when you have several virtual hosts, it is much easier to manage them when the configuration files are clearly descriptive.

RIT | Golisano College of Computing and Information Sciences School of Information

Please Note: The “*ServerAdmin*” statement is not necessary, it is purely an exercise in simulating a real-world entry, however, the other entries are required for the lab.

```
<VirtualHost *:80>
    ServerAdmin grock@gpavks.com
    DocumentRoot /www/virtualhosts/starlord.gpavks.com
    ServerName starlord.gpavks.com
    ErrorLog logs/starlord.gpavks.com-error.log
</VirtualHost>
```

See Figure 5 for a sample entry. Below is a summary of each statement in the file.

- The VirtualHost directive identifies the IP and port. The asterisk symbol, allows access through all interfaces followed by a colon and the HTTP well-known port, port 80.
- The ServerAdmin statement provides the email address of the administrator responsible for the site.
- The DocumentRoot statement is the location of the index.html file for the site.
- The ServerName statement is the virtual host server name for the site.
- The ErrorLog statement is where the error logs for each site are located.

Figure 5 – Example Virtual Host Entry

A screenshot of a terminal window titled "starlord.geenet.com.conf" [readonly] 8L, 212C. The window contains the following configuration code:

```
<VirtualHost *:80>
    ServerAdmin webmaster@starlord.geenet.com
    DocumentRoot /www/virtualhosts/starlord.geenet.com
    ServerName starlord.geenet.com
    ErrorLog logs/starlord.geenet.com-error_log
</VirtualHost>
```

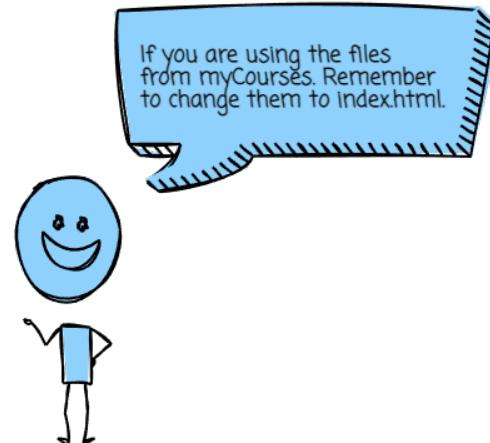
The terminal window has a menu bar with File, Edit, View, Search, Terminal, and Help. The status bar at the bottom right shows 8,0-1 and All.

- c. Create the directory path to point to the *index.html* file for the virtual host created for the site. In the following example, the path is */www/virtualhosts/starlord.gpavks.com/*.

```
$ mkdir -p /www/virtualhosts/starlord.gpavks.com
```

- d. Navigate to the `/www/virtualhosts/starlord.gpavks.com` directory and create a simple `index.html` file for testing purposes, or use the files provided in myCourses.
- e. Restart the `httpd` service and verify that the virtual host works.
- f. Repeat steps, b through e, for the second virtual host. The instructions use `gamora.gpavks.com` for the second virtual host.
- g. Finally, we need to create the default site, after all we want a default site in case something goes wrong with the virtual hosts. To do this, create a third configuration file in the `/etc/httpd/conf.d` directory. The file name must be, `"_default_.conf"`. No typos, it is underscore, followed by "default, and another underscore.
- h. The statements in the directive are similar to the virtual hosts; except that the `DocumentRoot` is `/var/www/html`.
- i. For a quick test to verify that the three sites work on the local server, you can add entries to the `/etc/hosts` file (see Figure 6 for an example). All the sites will point to server IP address, or you can use the loopback.

Figure 6 – Example Hosts Configuration File



```
student@localhost:~$ cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6

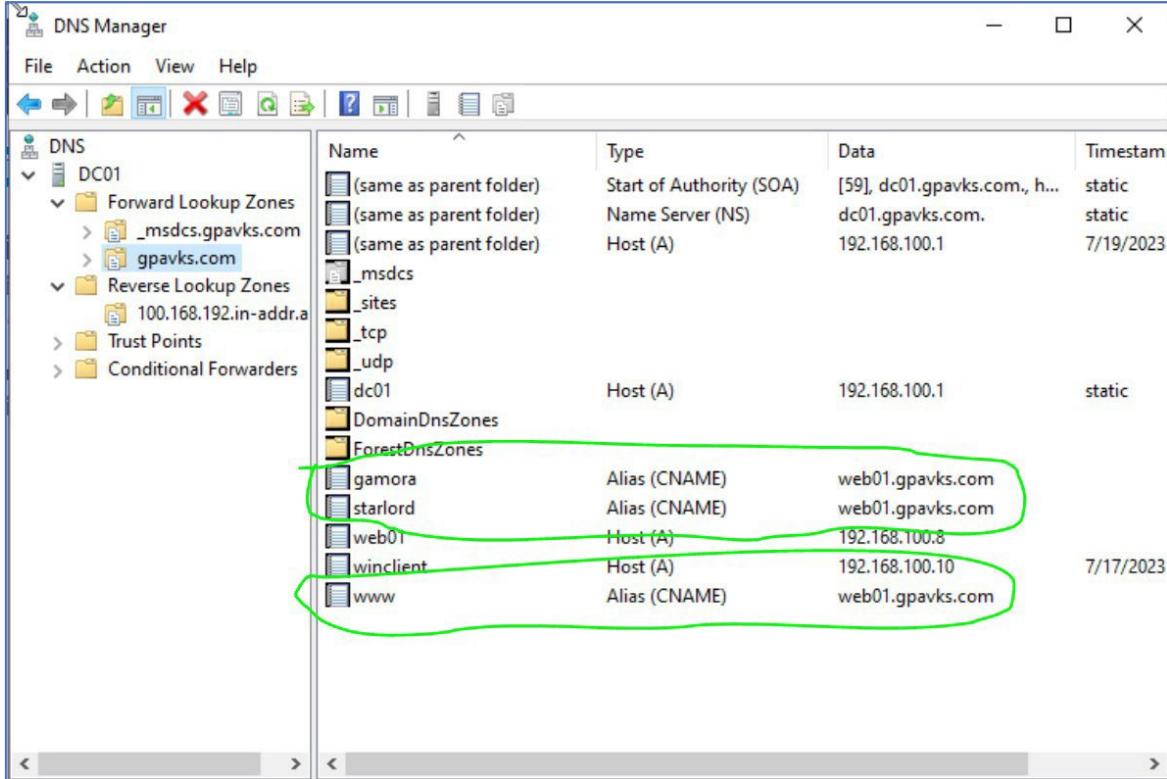
127.0.0.1      starlord.gpavks.com
127.0.0.1      gamora.gpavks.com
127.0.0.1      www.gpavks.com
[student@web01 ~]$
```

- j. Ideally, we want to access the sites remotely. First we need to create a firewall rule to allow incoming HTTP traffic on the web server. And then reload firewalld.

```
$ firewall-cmd --zone=public --add-service=http --permanent
$ firewall-cmd --reload
```

- k. Next, on Windows Server 2022, create aliases on the DNS server for each site. To do this, go to the Server Manager Dashboard and select *Tools → DNS* to launch the DNS Manager (Figure 7). Notice that three aliases are created for the two virtual hosts and one for the default. This is done by creating CNAME resource records.

Figure 7 – DNS Manager



DNS Configuration Recap

- Create a host record to map the hostname of the web server to its associated IP address (1). This may have been done in Lab 2. Referring to Figure 7, hostname web01 maps to the IP address 192.168.100.8.
- Map the default website using “www” as the DNS prefix (2), so when someone types in www.yourdomain.com, it directs them to the default website. This is mapped to the FQDN of the web server.
- Map the two virtual sites to their respective DNS prefixes (3 & 4), starlord and gamora respectively. Again, both of these are mapped to the FQDN of the web server.
- Using Figure 7 as an example, the default site is www.gpavks.com, and the other sites, starlord.gpavks.com and gamora.gpavks.com are associated to their respective sites but they all resolve to the FQDN of the web server, and the FQDN of the web server maps to its IP address.



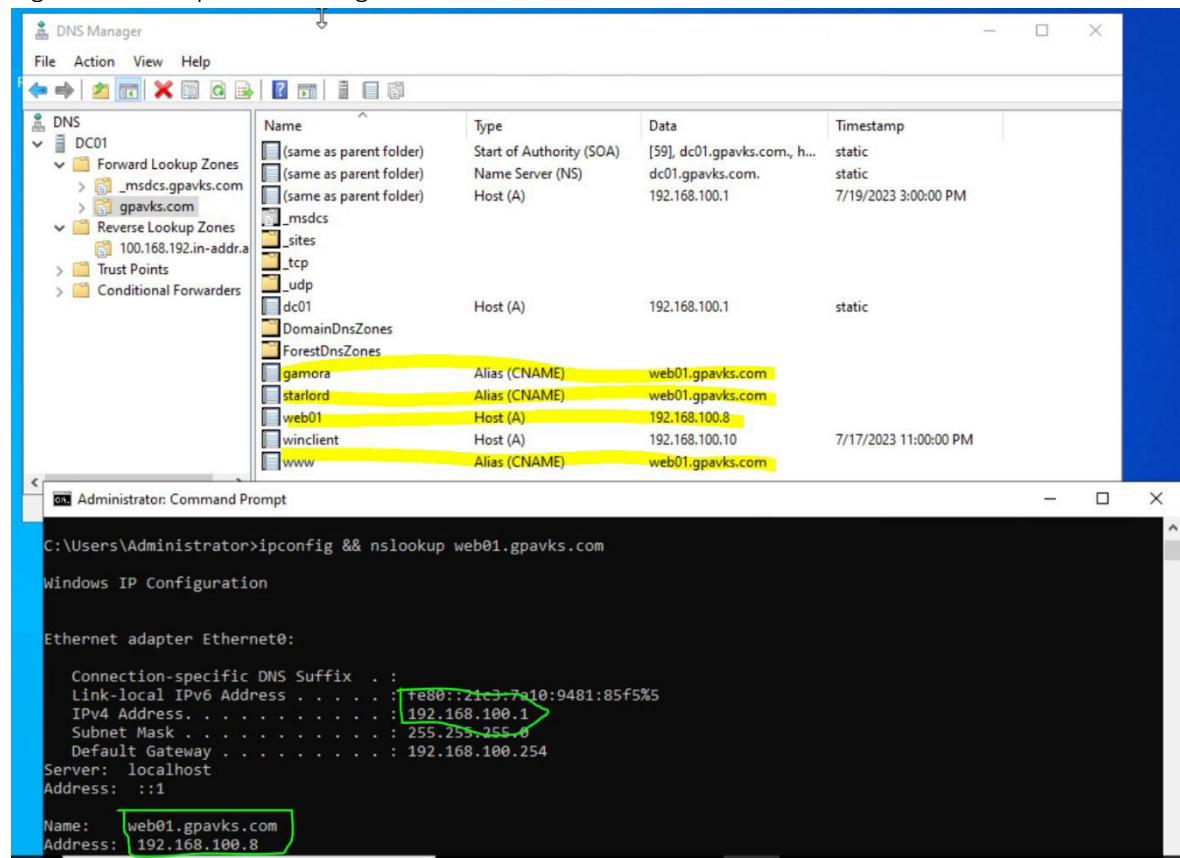
RIT | Golisano College of Computing and Information Sciences School of Information

For the report, include four screenshots.

1. The DNS configuration from Windows Server 2022. The screenshot must show the correct configurations for the following.
 - a. The A resource record that maps the Fully Qualified Domain Name (FQDN) or the web server.
 - b. The CNAME resource record that maps the default site to the FQDN of the web server.
 - c. The CNAME resource record that maps one of the virtual hosts to the FQDN of the web server.
 - d. The CNAME resource record that maps the other virtual host to the FQDN of the web server.

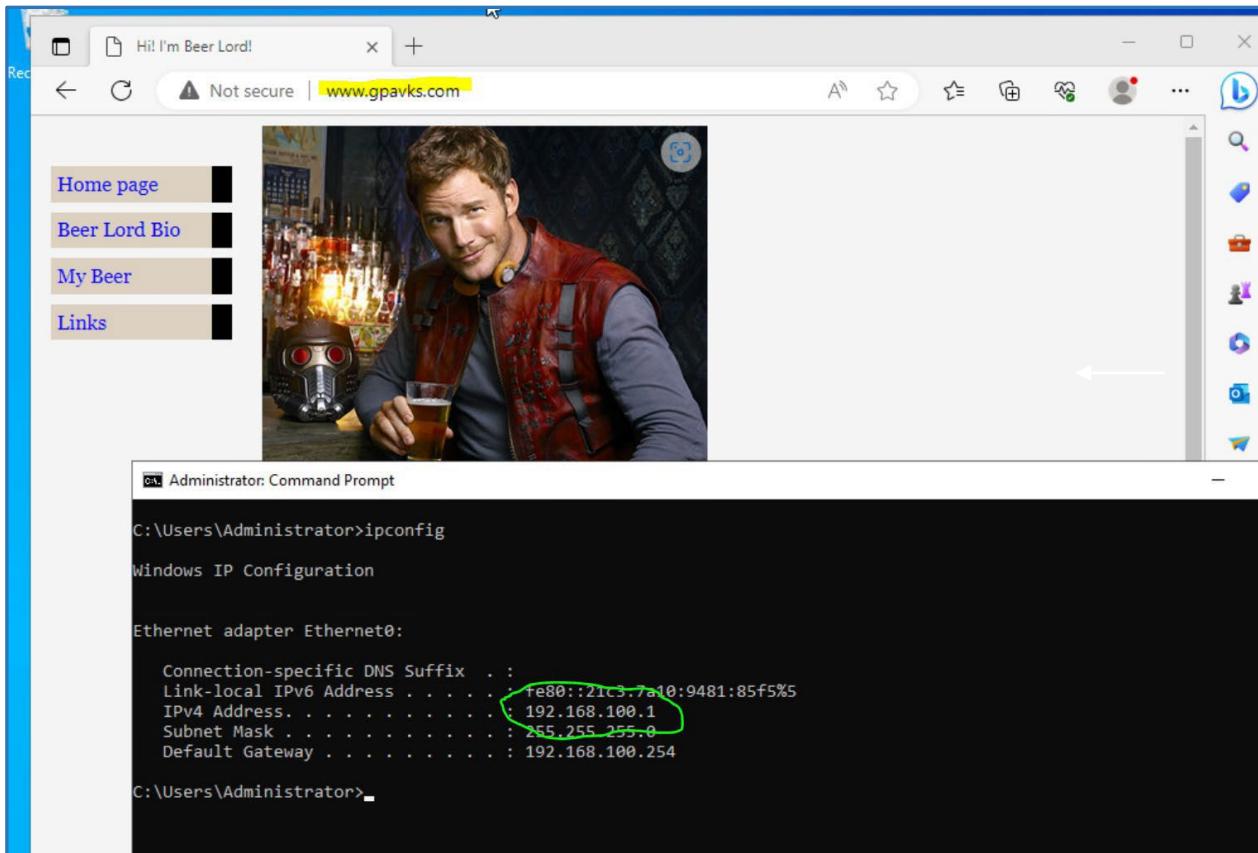
To validate the DNS CNAME resource record configuration, you must include a screenshot showing the records as they appear in the DNS manager. In the same screenshot, use the Windows Command Prompt, and show the output from the `ipconfig` command and `nslookup` on the web server. Refer to Figure 13 for an example.

Figure 8 – Example DNS Configuration



2. The second screenshot must show that the default site loads correctly from Windows Server 2022 using the default URL. The single screenshot must include the output from the `ipconfig` command, along with the web page; see Figure 9 for an example.

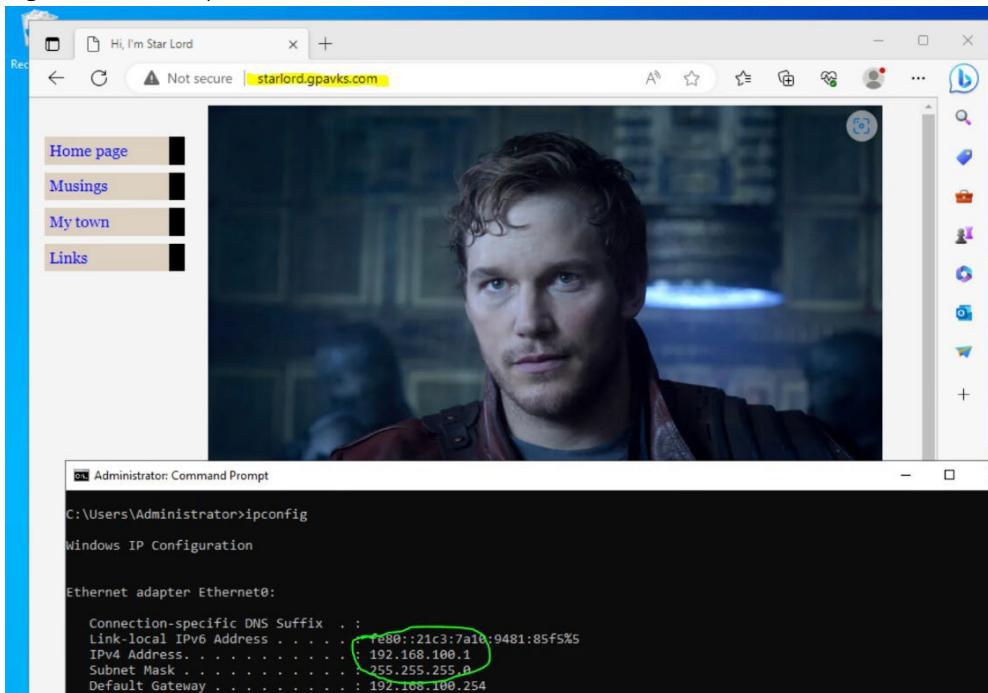
Figure 9 – Default Site



3. The third screenshot must show that the first virtual host loads correctly from Windows Server 2022 using the associative URL. The single screenshot must include the output from the `ipconfig` command, along with the web page; see Figure 15 for an example.

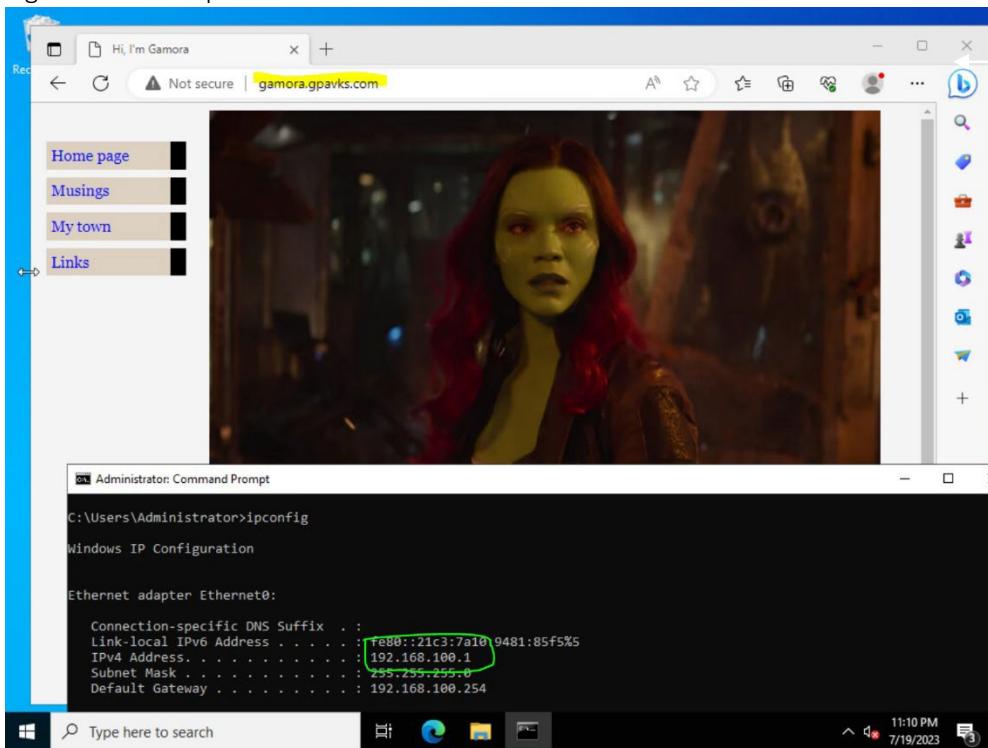
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 10 – Example for the First Virtual Host



4. The fourth screenshot must show that the site for the second virtual host loads correctly from Windows Server 2016 using the associative URL. The single screenshot must include the output from the **ipconfig** command, along with the web page; see Figure 11 for an example.

Figure 11 – Example for the Second Virtual Host



Activity 3 – Creating a Self-Signed Certificate

The next two activities focus on securing the web server. For this activity, you will create a self-signed certificate. The client uses the certificate to validate the authenticity of the web server and encrypt HTTP communications between itself and the server. Activity 4 will configure Apache to use Transport Layer Security.

- Log into the web server as root.
- As a reminder, SELinux can be disabled or set to permissive mode.
- You may need to install OpenSSL, depending on the version of Rocky it may already be installed, you can check with the following command, `rpm -qa openssl`. Otherwise, install using the following command.

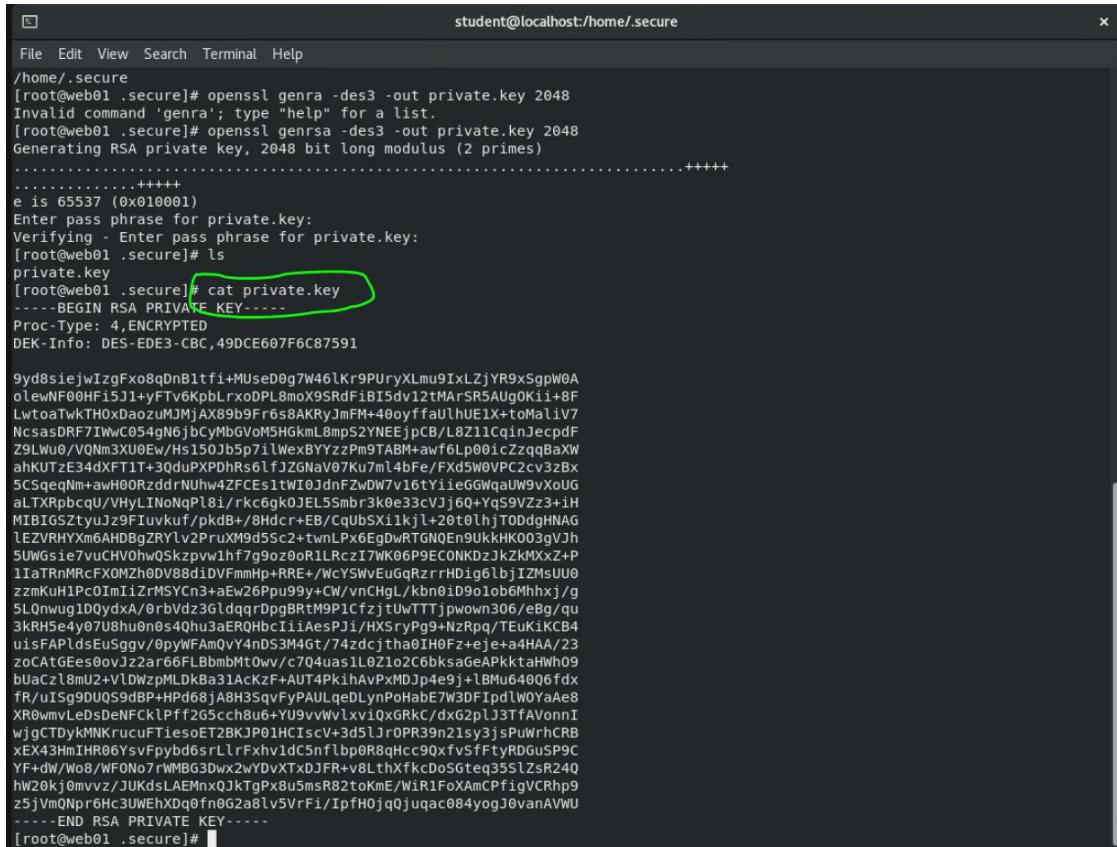
```
$ dnf install openssl -y
```

- Next, as root, create the private key using the following command. Be mindful of what directory you are in when the key is created, because that is the directory that it will reside. These instructions use the directory `/etc/pki/tls/private`.

```
$ openssl genrsa -des3 -out private.key 2048
```

- You will be prompted for a pass phrase for the key. Enter a passphrase, and commit to memory!
- To view the key, use the `cat` command.

Figure 12 – Example RSA Private Key



A screenshot of a terminal window titled "student@localhost:/home/.secure". The terminal shows the following command sequence:

```

File Edit View Search Terminal Help
student@localhost:/home/.secure
/home/.secure
[root@web01 .secure]# openssl genrsa -des3 -out private.key 2048
Invalid command 'genrsa'; type "help" for a list.
[root@web01 .secure]# openssl genrsa -des3 -out private.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
e is 65537 (0x010001)
Enter pass phrase for private.key:
Verifying - Enter pass phrase for private.key:
[root@web01 .secure]# ls
private.key
[root@web01 .secure]# cat private.key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,49DC607F6C87591

9yd8siejwIzgFxo8qDnBltfi+MuSeD0g7W46lKr9PUryXlmu9IxLzjYR9xSgpW0A
oleWF0#Hf15J1+yFtvGpkBLrxoDPLBmox95RdfiB15vd1zMar5R5AuGOkII+8f
LwtoatTwkTH0xDaozuUMMjAX89b9Fr6s8AKRyJmfM=40oyffaUlhUE1x+toMalv7
NcsasDRF7IwxC054gN6jbCyMb6V0m5HGkmL8mpS2YNEEjpCB/L8Z11CqinJecpdF
Z9LWu0/VQNm3XU0Ew/Hs150jb5p7ilWexByzzPm9TABM+awfLp00icZsqqBaXW
ahkUTz34dXFT1T+30duXPXPDRhs6lfjZGNva07Ku7ml4bFe/Fxd5w0VPC2cv3zbX
5CSqeQNm+awh0ORzddrNuHw4ZFCes7WI0JdnFzW7v16TYiieGGWqaUw9vxouUG
aLTXRpbccU/HyINoNpLbji/rkc6qkOJEL5Smbr3k0e33cVj6Q+Yqs9Vz3+iH
MIBIGSztyJz9FiuVkuF/pkDB+/8Hdcr+E/B/cuObSXii1kj1+20t0lhjT0DdgHnAG
LEZVRHYxM6AHDbgZRYlv2PruXm9d5C2+twLPx6EgDwRTGQNEn9ukKKHO03gVjh
5UWGsiEesovvuCHvOhwQskzpvw1hf7g9oz0o11RczI7Wk06P9ECONKDzJkZkMxxZ+p
11aTRnMrcFXOMZh0DV88d1DVfmMp+RRE+/WcYSWvEuGqrzrHDig6bjj1ZMsUU0
zzmKuHPcoImIizrMSYCN3+aEW26pu99y+Cw/vnChgl/kbn0D9o1ob6mhxj/g
5L0nwug1DqdydxA/0rbVdz3lddqfDpgBRtM9P1cfzjtUwTTTjpwown306/eBg/qu
3KRh5e4y07U8hu0nes40hu3aER0Hbcii1aesJi/HXsryPg9+NzRpq/TEuK1kCB4
uisFAPLdsEuSggv0pywfAmQvY4nDS3M4Gt/74zdjcjta0Tl0Fz+ej+a4HAA/23
zoCAtGEesovvJzkar6fLBbm0Mt0w/c704uas1l0z1o2c6bsageApKktarWh09
buaCz18mU2+vL0wzpMLdkBa31ackzf-AUT4PkihAvPxMDjP4e9j+LMu6400fwdx
fr/u1Sg90uQ59dBP+HPd68jA8H3SqvFyDULqebDlynPoHabE7W3DFIpdlw0YaaE8
XR0wmwLeDsDeNFCKlPff265cc8u6+YU9vvWlxviQxGRKC/dxG2plj3TfAVonnI
wjgCTDykMNkrucuFTiesoT2BKJp01Hc1cs+3d5lrlpr39n21sy3jsPuWrHCRB
eX43HmIHr06YsvFpybd6sL1rfxhv1dcCnf1bp0R8qHcc90xfv5FtyRDGusP9C
YF+dW/W08/wFONo7rwMBG3Dwx2wDVxtxDJFR+v8LthxFkcDo5teq35SLzsr240
hW20kj0mvvz/JUKdsLAEMnxQJKtgPx8u5msR82toKmE/WiR1FoXAmCPfigVCRhP9
25VmQnp6Hc3UWehxDq0fn0G2a8lv5Vrfi/IptH0jqQjuqac084yogJ0vanAvWU
-----END RSA PRIVATE KEY-----
[root@web01 .secure]#

```

- Next, create the Certificate Signing Request (CSR) which is used to (you guessed it), sign your certificate. When prompted enter the following information. These instruction store the

```
$ openssl req -key private.key -new -out server.csr
```

- Country: US
- State: New York
- Locality: Rochester
- Company: RIT
- Organizational Unit: NSSA221
- FQDN of the web server: **web01.gpavks.com**
- Email: me@me.com
- Challenge Password: <you decide>
- Optional Company Name: RIT

For our purposes, most of the information is unimportant. However, you must enter the FQDN for your server.

Figure 13 – Sample CSR Configuration

```
student@localhost:/home/.secure
File Edit View Search Terminal Help
[root@web01 .secure]# openssl req -key private.key -new -out server.csr
Enter pass phrase for private.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) [Default City]:Rochester
Organization Name (eg, company) [Default Company Ltd]:RIT
Organizational Unit Name (eg, section) []:NSSA221
Common Name (eg, your name or your server's hostname) []:web01.gpavks.com
Email Address []:gpavks@rit.edu

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:eyeofrror
An optional company name []:RIT
[root@web01 .secure]#
```

- Next, create the Self-Signed Certificate. This certificate will be used to encrypt the HTTP traffic. Using the CSR and private key created in the previous steps enter the following command. These instructions create the certificate in the /etc/pki/tls/certs directory. Please note that the command is a single line.

```
$ openssl x509 -signkey /etc/pki/tls/private/private.key -in
/etc/pki/tls/private/server.csr -req -days 365 -out server.crt
```

- Upon completion of the command, you will be prompted to enter the pass phrase for the private key (hope you remembered it!). Once complete the output will look similar to Figure 14.

Figure 14 – Certificate Creation

```
[root@web01 .secure]# openssl x509 -signkey private.key -in server.csr -req -days 365 -out server.crt
Signature ok
subject=C = US, ST = New York, L = Rochester, O = RIT, OU = NSSA221, CN = web01.gpavks.com, emailAddress = gpavks@rit.edu
Getting Private key
Enter pass phrase for private.key:
[root@web01 .secure]#
```



For the report, use the following command to show the contents of the certificate in plain text, `openssl x509 -text -noout -in server.crt`. Make sure to include the `hostname` and `date`. Figure 15 provides sample output.

Figure 15 – Certificate Verification Example

```
student@localhost:/home/secure
File Edit View Search Terminal Help
[root@web01 .secure]# hostname; date; openssl x509 -text -noout -in server.crt
web01.gpavks.com
Thu Jul 20 08:55:02 EDT 2023
Certificate:
Data:
    Version: 1 (0x0)
    Serial Number:
        16:52:1e:b9:11:ce:03:36:34:13:4b:7a:68:c9:bf:93:e2:9c:e7:7c
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = US, ST = New York, L = Rochester, O = RIT, OU = NSSA221, CN = web01.gpavks.com, emailAddress = gpavks@rit.edu
    Validity
        Not Before: Jul 20 12:44:26 2023 GMT
        Not After : Jul 19 12:44:26 2024 GMT
    Subject: C = US, ST = New York, L = Rochester, O = RIT, OU = NSSA221, CN = web01.gpavks.com, emailAddress = gpavks@rit.edu
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            RSA Public-Key: (2048 bit)
                Modulus:
                    00:df:58:00:e7:ed:4e:f0:bf:24:86:4c:0c:4f:29:
                    82:da:17:3b:fb:de:a3:bf:2d:da:ca:5c:32:a4:c5:
                    88:5f:04:c9:8e:ce:48:b2:62:0f:ee:ce:a1:9f:a3:
                    60:8e:1f:af:1c:ad:da:94:ac:74:21:8a:50:b1:d9:
                    87:84:08:11:e0:a0:c5:fa:dd:bd:58:bc:5d:26:
                    61:33:eb:59:99:0e:ab:ee:4e:41:81:7c:08:3f:7f:
                    ed:0b:6c:6e:db:da:cd:d2:5e:84:6d:30:f6:0c:4a:
                    17:f4:4b:b9:97:8c:a5:6a:85:19:55:e4:7b:ff:9b:
                    8f:08:cb:11:d7:db:6c:80:bd:b5:3e:23:8c:1a:b1:
                    36:f2:af:ed:b9:f3:4b:d6:e6:b6:a5:55:22:9d:77:
                    af:e9:51:f6:a3:1b:c0:09:54:5a:ed:a9:46:44:99:
                    c1:1a:ac:27:61:61:07:f0:12:65:36:36:5c:96:5d:
                    e8:e7:41:08:dd:43:a5:83:76:bf:9d:93:fcc:7e:aa:
                    26:6c:e2:64:af:ce:83:71:f7:45:f9:5c:00:c9:4b:
                    a9:0b:09:41:e8:c:ba:22:f3:e6:f:fa:a9:a9:d8:
                    13:84:7b:dd:d7:9c:ab:ab:97:94:a9:9a:aa:7a:8c:
                    99:e2:3d:fd:ee:46:82:4d:a5:f8:b8:b3:d4:b5:46:
                    9a:5d
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
        11:a4:02:c1:1:85:a6:c5:be:ae:f3:d2:c8:ad:6e:9f:a0:5a:
        42:0f:b8:6c:3e:03:e6:72:31:14:63:76:a6:41:04:15:ae:54:
        dd:b7:e5:6b:27:ed:dd:7:c3:e9:1e:80:dc:22:b0:e4:a2:dc:
        f8:ad:f3:70:f5:c8:ec:f2:23:4b:9e:23:31:50:03:88:b2:73:
        4b:b1:56:91:25:67:58:8f:52:b1:db:0b:32:c1:93:46:2d:ae:
        9a:01:10:89:2e:96:ba:31:8c:fc:02:5a:ee:96:8f:e7:74:98:
        89:c4:19:99:07:8c:2b:ec:7e:e3:0c:e2:89:68:e3:99:59:a7:
        b0:bc:7f:05:a6:82:69:60:47:64:3e:19:e1:27:a2:c6:e5:
        52:78:ee:76:c7:e5:e8:42:fb:75:27:09:37:55:1c:60:48:37:
        46:52:27:9e:bf:69:a5:2c:8e:68:83:b3:a9:9f:6c:b1:c5:
        d6:4c:82:05:4d:af:f5:18:78:55:30:c3:6a:12:34:71:c5:f1:
        f6:0e:26:5b:5b:2e:92:b9:a3:32:24:be:bb:cb:14:c1:6d:11:
        97:d4:da:1f:08:19:58:1f:d7:26:11:d7:b9:9f:73:4c:f1:88:
        96:53:19:01:de:32:eb:0f:fb:fa:8f:a0:84:d6:24:2b:f6:7c:
        ab:e2:a7:07
[root@web01 .secure]#
```

Activity 4 – Configuring Apache for Transport Layer Security

Now that we have the certificate, the next step is to configure the Apache server for TLS. All examples in this activity will use the following FQDN for the web server, `web01.gpavks.com`.

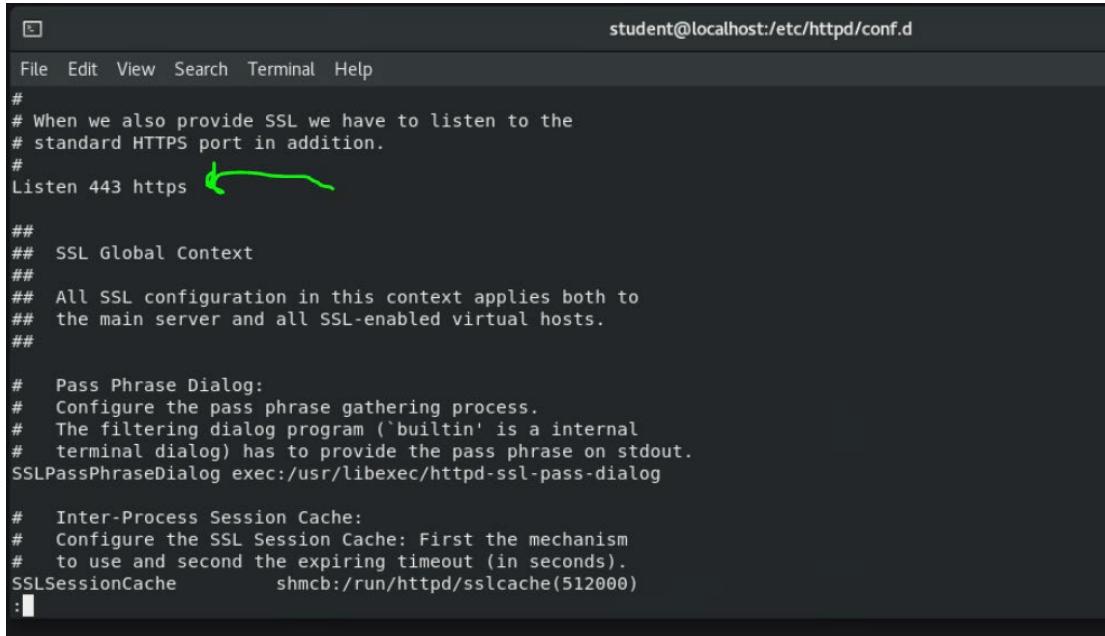
- First, we need to install the required Apache modules.

```
$ dnf -y install mod_ssl
```

RIT | Golisano College of Computing and Information Sciences School of Information

- b. Navigate to the `/etc/httpd/conf.d` directory and locate the `ssl.conf` file. This file contains the information needed to create the secure virtual host. Before moving on to the next steps look at some of the statements in the file that are relevant to the exercise (steps c through e).
- c. Figure 16 shows the directive that uses port 443 for HTTPS.

Figure 16 – SSL Port



```
student@localhost:/etc/httpd/conf.d

File Edit View Search Terminal Help
#
# When we also provide SSL we have to listen to the
# standard HTTPS port in addition.
#
Listen 443 https ↗

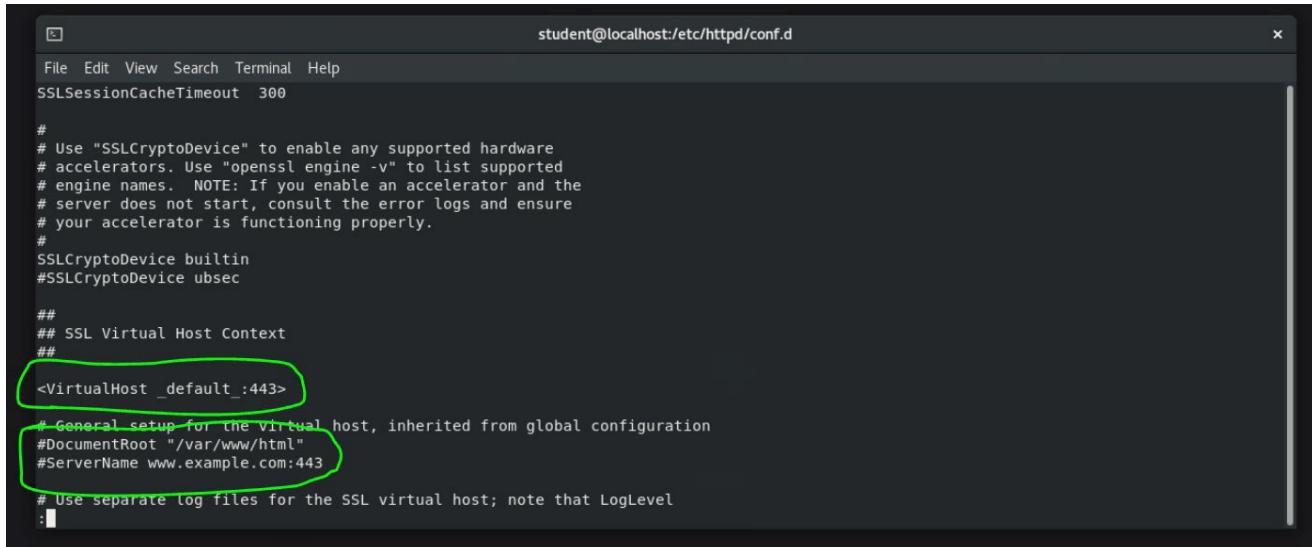
##
## SSL Global Context
##
## All SSL configuration in this context applies both to
## the main server and all SSL-enabled virtual hosts.
##

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program ('builtin' is a internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use and second the expiring timeout (in seconds).
SSLSessionCache      shmcb:/run/httpd/sslcache(512000)
:|
```

- d. Figure 17 shows the statements that define the default virtual host configuration settings; we do not need to change the information because it is already set up in the global configuration of the `httpd.conf` file. For this activity, you will use the default virtual host from the previous exercise whose `index.html` file is located in `/var/www/html`.

Figure 17 – Default Virtual Host



```
student@localhost:/etc/httpd/conf.d

File Edit View Search Terminal Help
SSLSessionCacheTimeout 300

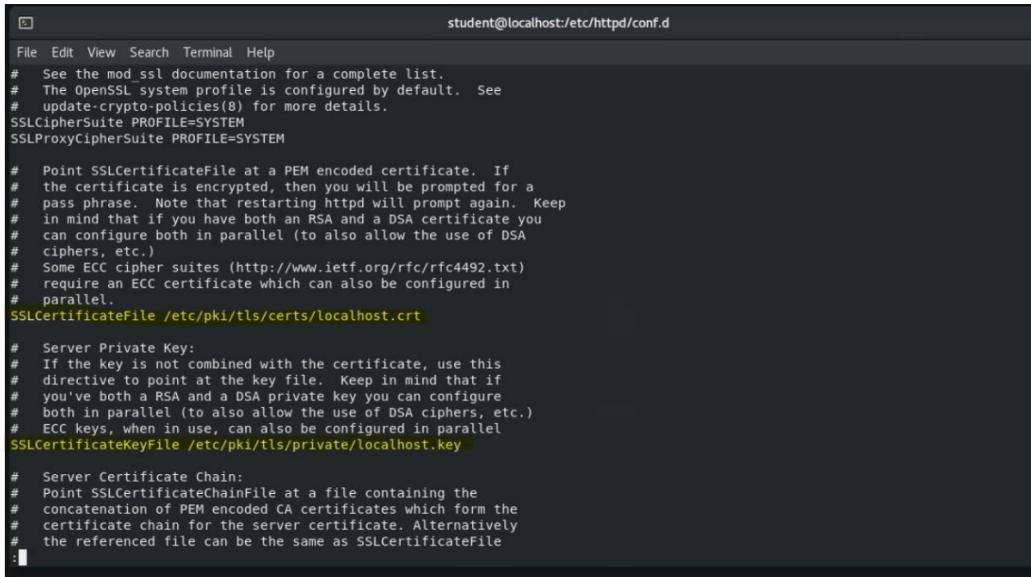
#
# Use "SSLCryptoDevice" to enable any supported hardware
# accelerators. Use "openssl engine -v" to list supported
# engine names. NOTE: If you enable an accelerator and the
# server does not start, consult the error logs and ensure
# your accelerator is functioning properly.
#
SSLCryptoDevice builtin
#SSLCryptoDevice ubsec

##
## SSL Virtual Host Context
##
<VirtualHost _default_:443>
    # General setup for the virtual host, inherited from global configuration
    #DocumentRoot "/var/www/html"
    #ServerName www.example.com:443
    # Use separate log files for the SSL virtual host; note that LogLevel
:|
```

RIT | Golisano College of Computing and Information Sciences School of Information

- e. Now we get to the important stuff. This section of the SSL configuration file tells the httpd daemon service the location of the private key and the self-signed certificate created in the previous activity. You will need to edit the file for the location of the key and certificate on your server. The screenshot shows the default locations.

Figure 18 – Certificate and Key Default Location



```
student@localhost:/etc/httpd/conf.d
File Edit View Search Terminal Help
# See the mod_ssl documentation for a complete list.
# The OpenSSL system profile is configured by default. See
# update-crypto-policies(8) for more details.
SSLCipherSuite PROFILE=SYSTEM
SSLProxyCipherSuite PROFILE=SYSTEM

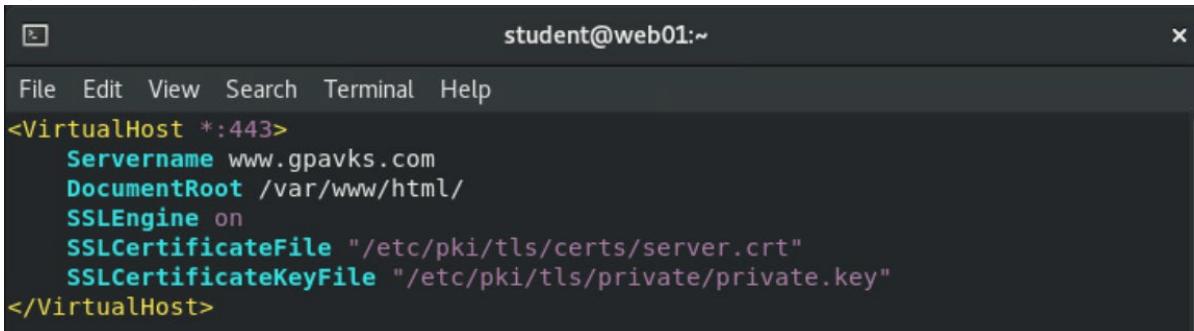
# Point SSLCertificateFile at a PEM encoded certificate. If
# the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that restarting httpd will prompt again. Keep
# in mind that if you have both an RSA and a DSA certificate you
# can configure both in parallel (to also allow the use of DSA
# ciphers, etc.)
# Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
# require an ECC certificate which can also be configured in
# parallel.
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
# ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
:
```

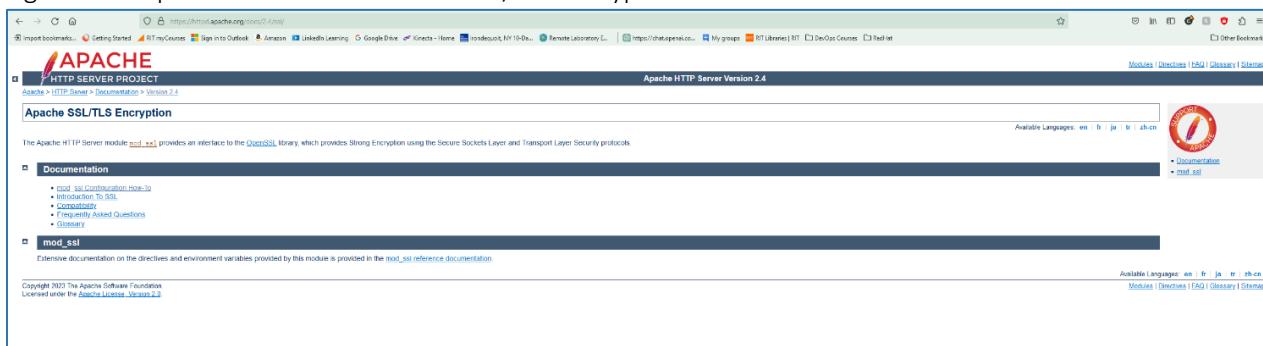
- f. Next, create the secure virtual host in the `/etc/httpd/conf.d` directory. Using a text editor create a file for the secure virtual host and give it a descriptive name, for example, “`webserver_ssl.yourid.com.conf`”. Figure 19 provides a sample configuration file; the statements should be familiar to you with the exception of the SSLEngine, SSLCertificateFile, and SSLCertificateKeyFile. As always, for more information you can refer to the Apache documentation.

Figure 19 – Sample Secure Virtual Host File



```
student@web01:~
File Edit View Search Terminal Help
<VirtualHost *:443>
    Servername www.gpavks.com
    DocumentRoot /var/www/html/
    SSLEngine on
    SSLCertificateFile "/etc/pki/tls/certs/server.crt"
    SSLCertificateKeyFile "/etc/pki/tls/private/private.key"
</VirtualHost>
```

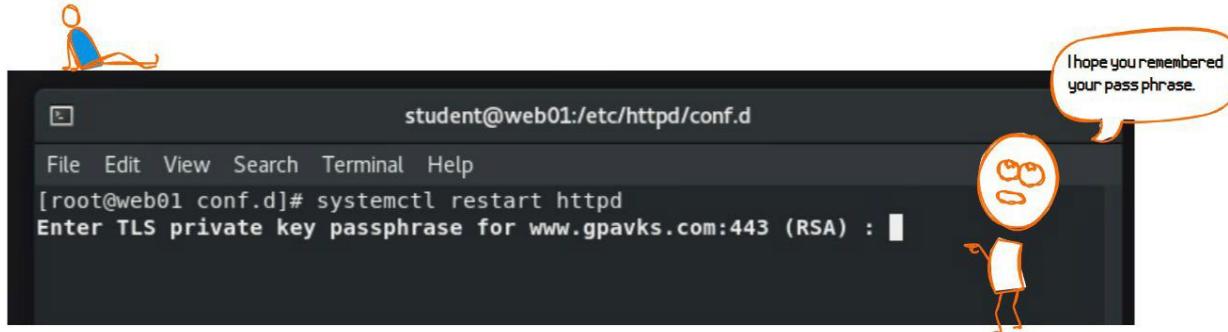
Figure 20 – Apache Documentation for SSL/TLS Encryption



The screenshot shows a web browser displaying the Apache HTTP Server Version 2.4 documentation for the mod_ssl module. The URL is <https://httpd.apache.org/docs/2.4/>. The page title is "Apache SSL/TLS Encryption". The content includes sections for "Documentation" (mod_ssl Configuration How-to, Introduction to SSL, Configuration Examples, Frequently Asked Questions, Glossary) and "mod_ssl" (Extensive documentation on the directives and environment variables provided by this module is provided in the mod_ssl reference documentation). The page footer indicates "Copyright 2023 The Apache Software Foundation. Licensed under the Apache License, Version 2.0".

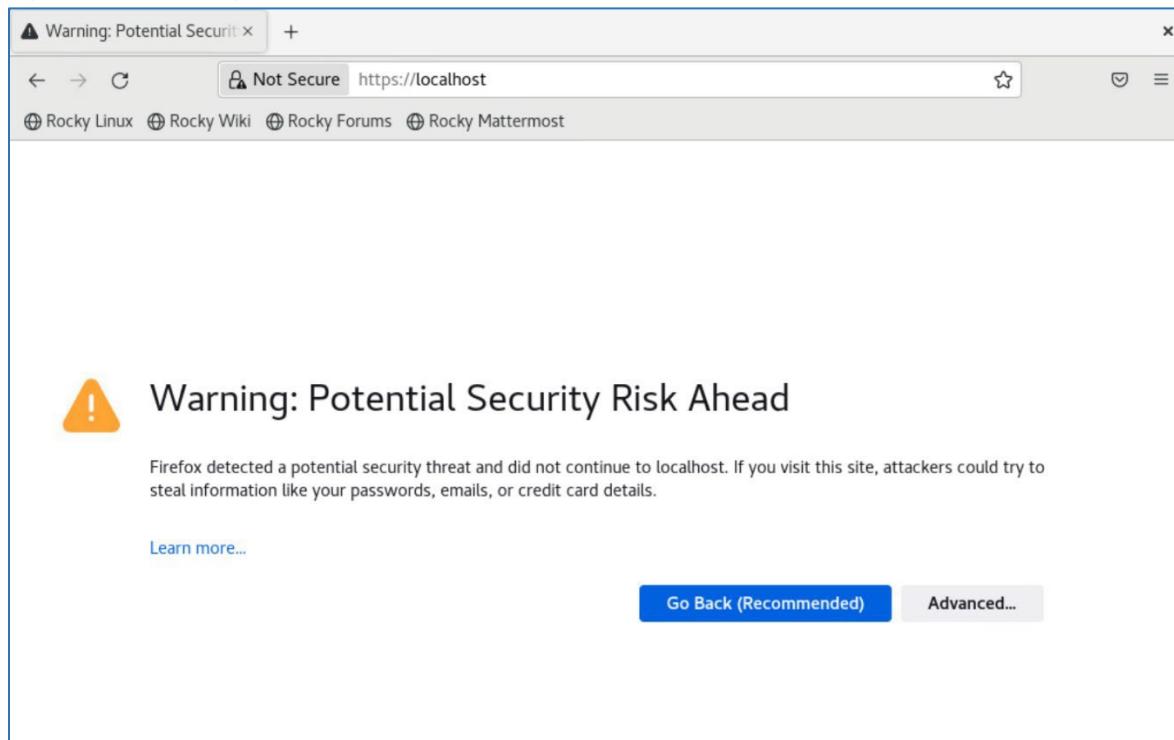
- g. Next, restart the http daemon process. Assuming you configured correctly a passphrase for the key will need to be entered when the service restarts.

Figure 21 – Enter Key Passphrase



- h. Using the browser on the local server, enter **https://localhost** ("s" in bold for emphasis). The site will inform you that the connection is a potential security risk (Figure 22).
- i. Click the "Advanced" button (Figure 22).

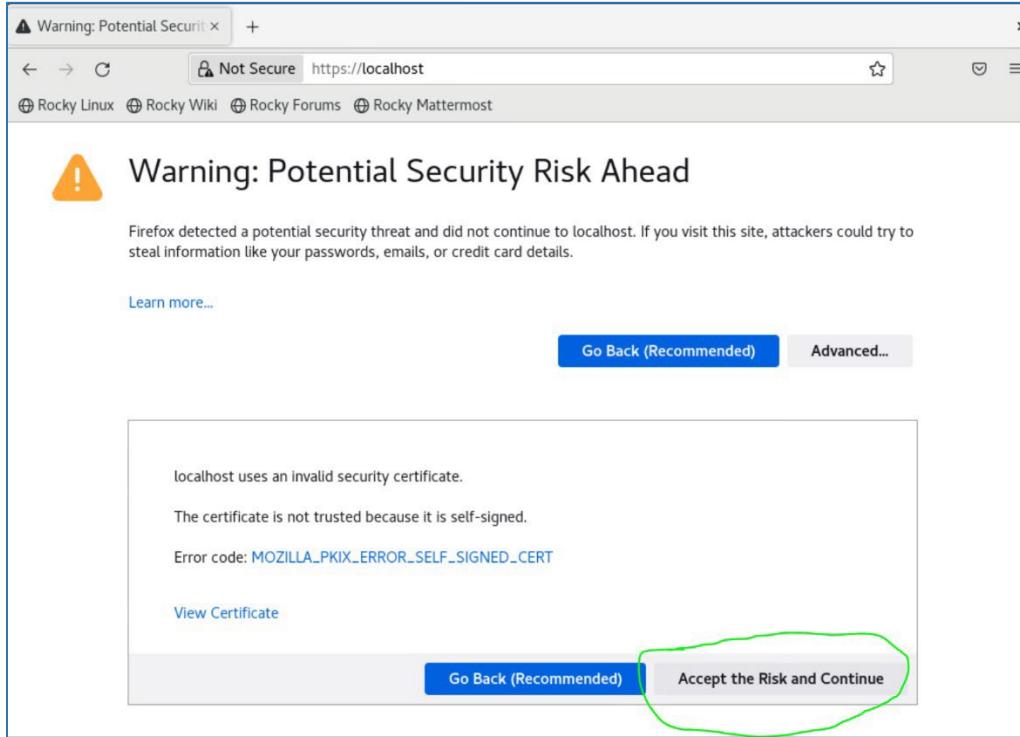
Figure 22 – Warning



- j. To retrieve the self-signed certificate from the server, click the "Accept the Risk and Continue" button (Figure 23).

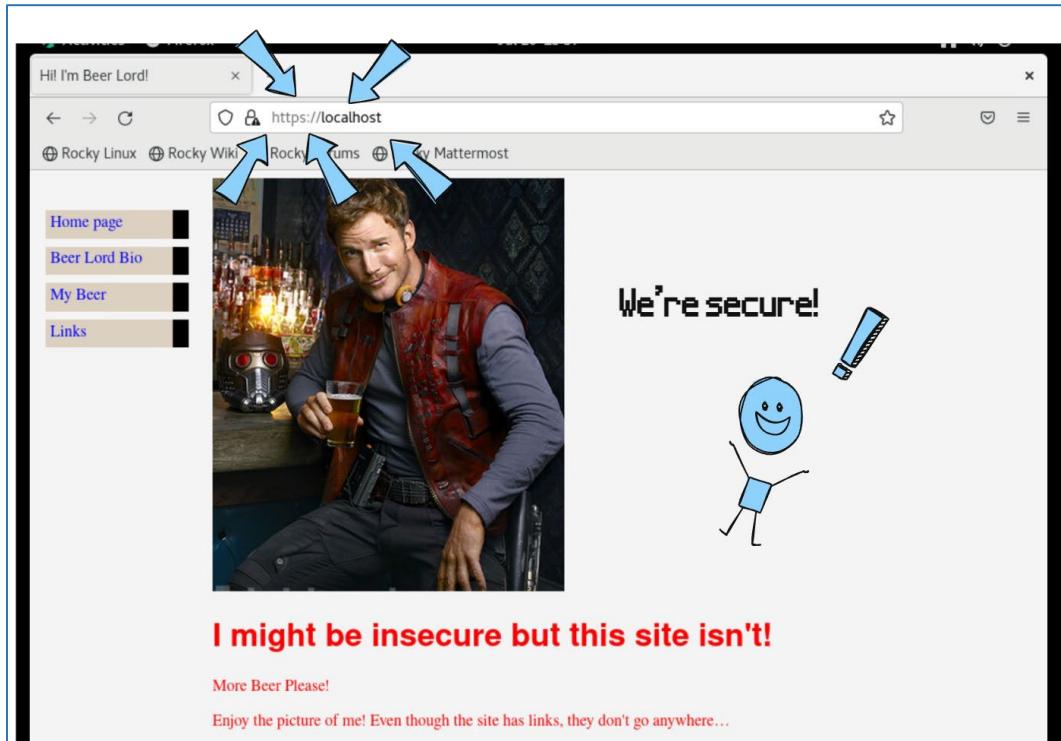
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 23 – Accept Risk



- k. This will launch the secure site (Figure 24).

Figure 24 – Secure Virtual Site





For the report, two screenshots are needed. The first screenshot must show that the secure site loads correctly (Figure 24). The second is a single screenshot showing your FQDN, and the output from the `hostname -I` command, and the `ls -l` command for the file that you created for the secure virtual host. In the same screenshot, use the `cat` command to show the contents of the file. The diagram must be a single screen shot properly labelled and included in the report. Refer to Figure 25 for an example.

Figure 25 – Example Screenshot for SSL Virtual Host Validation

The screenshot shows a terminal window titled "student@web01:/etc/httpd/conf.d". The window contains the following text:

```
File Edit View Search Terminal Help
[root@web01 conf.d]# hostname; hostname -I; ls -l webserver_ssl.gpavks.com.conf
web01.gpavks.com
192.168.100.8 192.168.122.1
-rw-r--r--. 1 root root 232 Jul 20 13:25 webserver_ssl.gpavks.com.conf
[root@web01 conf.d]# cat webserver_ssl.gpavks.com.conf
<VirtualHost *:443>
    Servername www.gpavks.com
    DocumentRoot /var/www/html/
    SSLEngine on
    SSLCertificateFile "/etc/pki/tls/certs/server.crt"
    SSLCertificateKeyFile "/etc/pki/tls/private/private.key"
</VirtualHost>

[root@web01 conf.d]#
```

Activity 5 – Installing and Configuring MailEnable

In this activity, you're tasked with setting up and configuring a mail server within your infrastructure. This process will involve using MailEnable, a free mail server software, which you'll install on Windows Server 2022. MailEnable will serve dual roles as both the Mail Delivery Agent (MDA) and the Mail Transfer Agent (MTA). This means it will handle both SMTP and IMAP traffic.

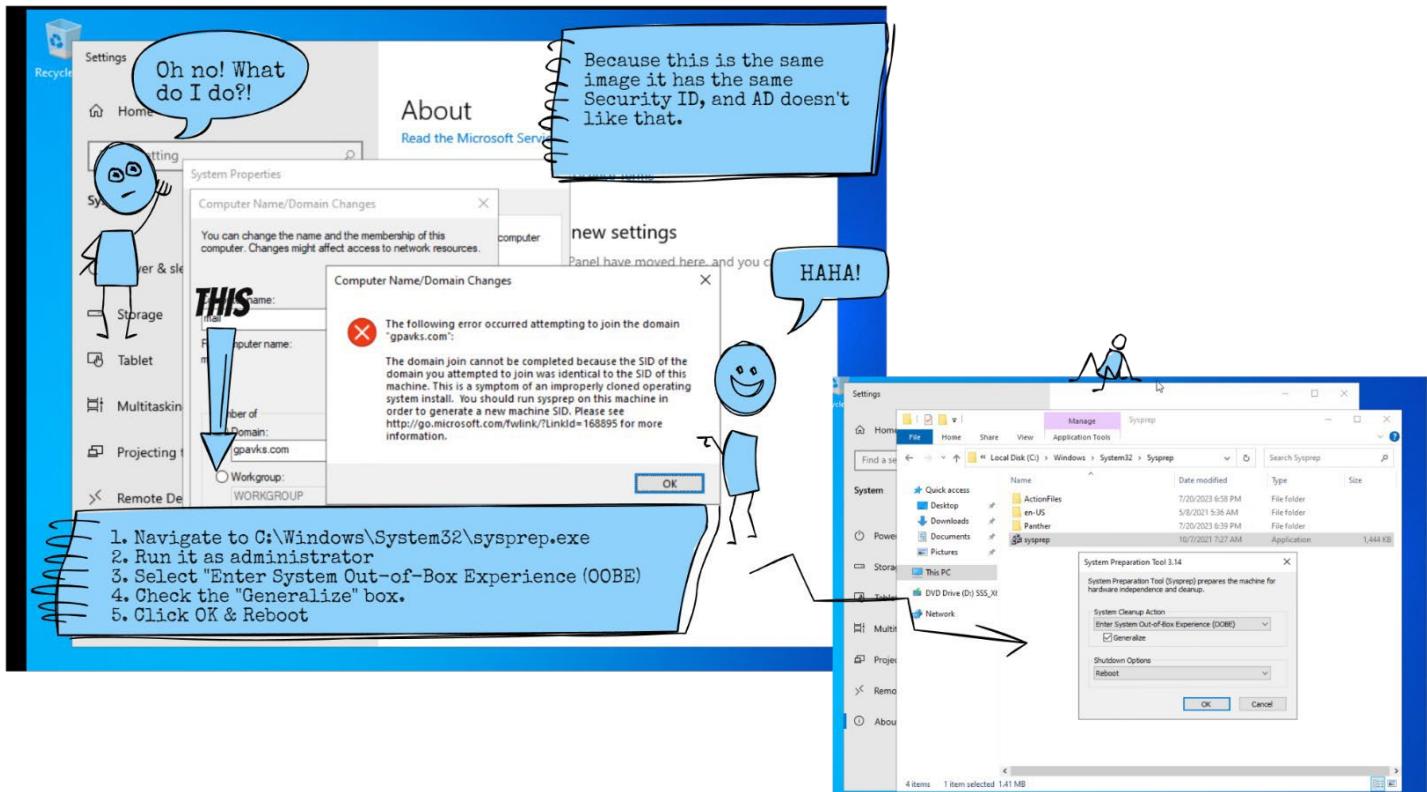
Subsequently, you'll need to install the Mail User Agent (MUA), which is the client software, on Windows 10 and Rocky Linux. Your task also includes creating two mailboxes and sending an email from one user to another.

Once you have everything set up and functional, your final task will be to capture and analyze email traffic. This process is crucial for understanding the active protocols and learning the intricacies of email communication.

- Deploy a new instance of Windows Server 2022. You will need to run the `sysprep` utility because it has the same Security ID as the Windows Server in your deployment (see Figure 26).

RIT | Golisano College of Computing and Information Sciences School of Information

Figure 26 – Sysprep Procedure



- b. After the system reboots, give it a meaningful hostname (i.e., something that will identify it as the mail server). Configure static network settings. Make sure to join the server to the domain using a domain admin account.
- c. Update your DNS with an MX and A records for the mail server.
- d. Using the “Add Roles and Features Wizard,” install Web Server (IIS) and add the default features. This is a prerequisite for MailEnable.
- e. Using the following URL – <https://www.mailenable.com>, download and install the free Standard Edition. You DO NOT need to register to download MailEnable.

A little review of
Labs 1 & 2.

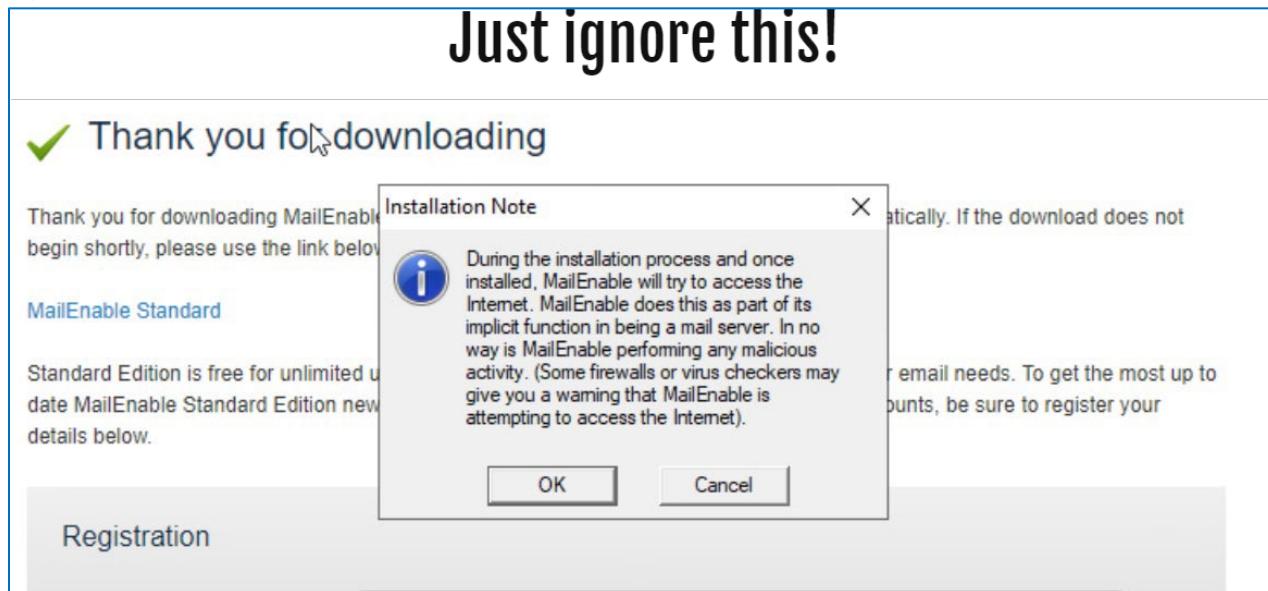
Figure 27 – MailEnable Standard Edition

The screenshot shows the MailEnable website homepage. The 'Standard Edition (FREE)' option is circled in green. Other editions listed are 'Professional Edition', 'Enterprise Edition', and 'Enterprise Premium'. The page also features sections for 'World's Most Popular Windows Mail Server', 'Microsoft Exchange Alternative', and 'Affordable Unlimited User Licensing'.

RIT | Golisano College of Computing and Information Sciences School of Information

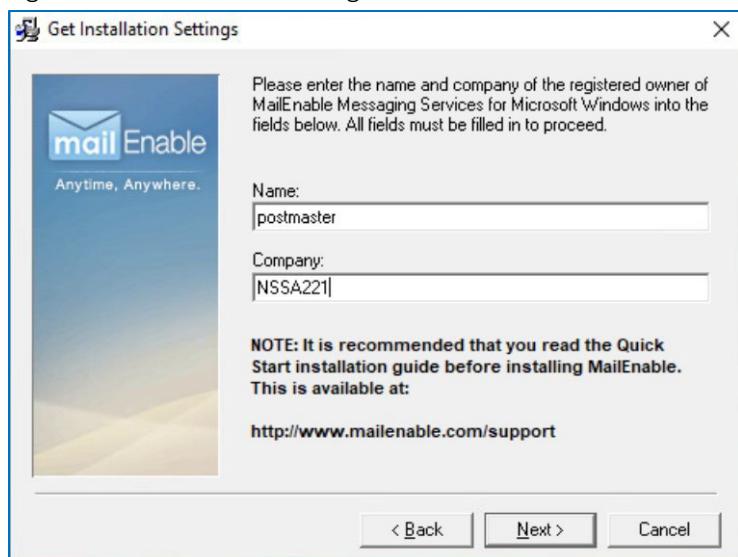
- f. During the installation, unless otherwise noted select the default settings.
- g. You can read the installation note (but you probably won't).

Figure 28 – Installation Note



- h. When prompted input a name and the company (make it up), and click Next.

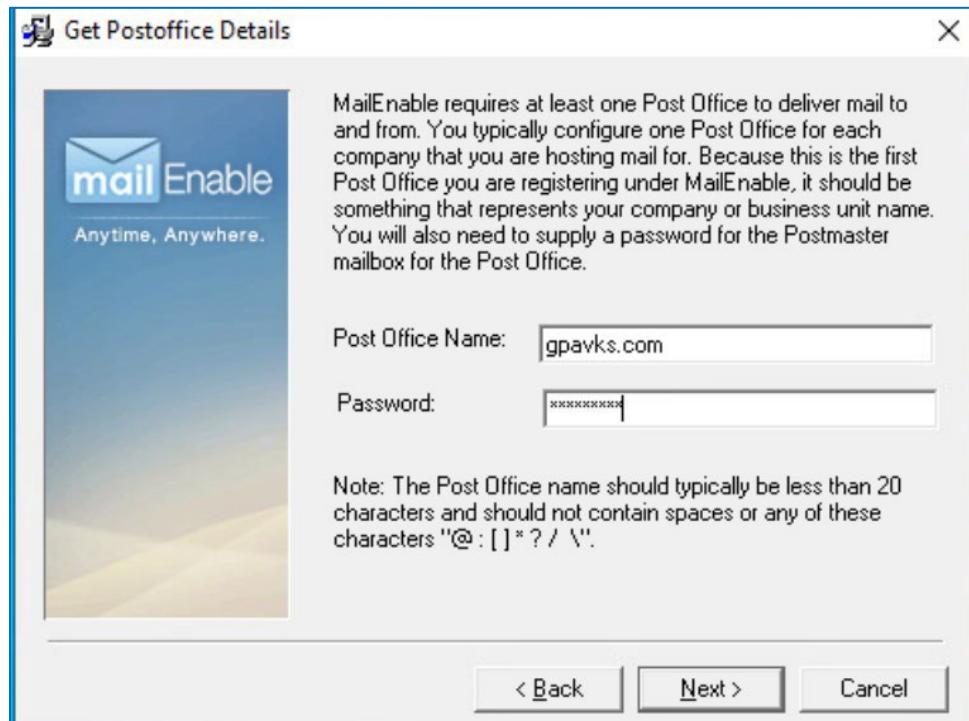
Figure 29 – Installation Settings



- i. Hit Next a bunch of times until you get to "Get Postoffice Details." The Post Office Name field will be populated with your domain (not mine, like Figure 30). And enter a password, then hit Next.

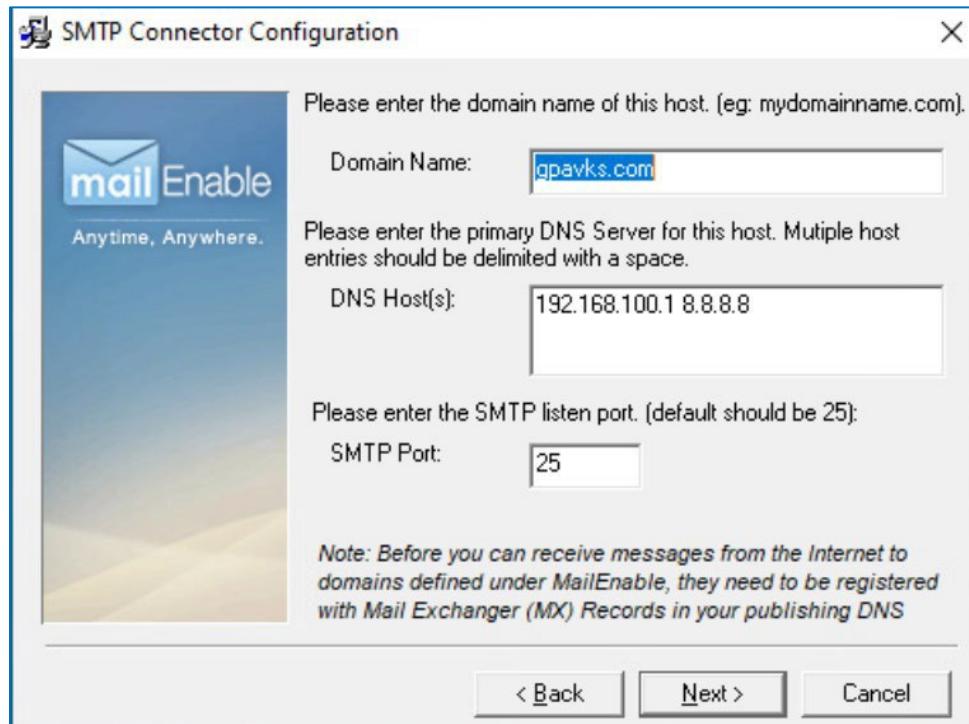
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 30 – Post Office Details



- j. Hit next a couple more times and verify that the information in the SMTP Connector Configuration is correct. Again, it should be your domain, and DNS information, not what's in the image (the image just for reference).

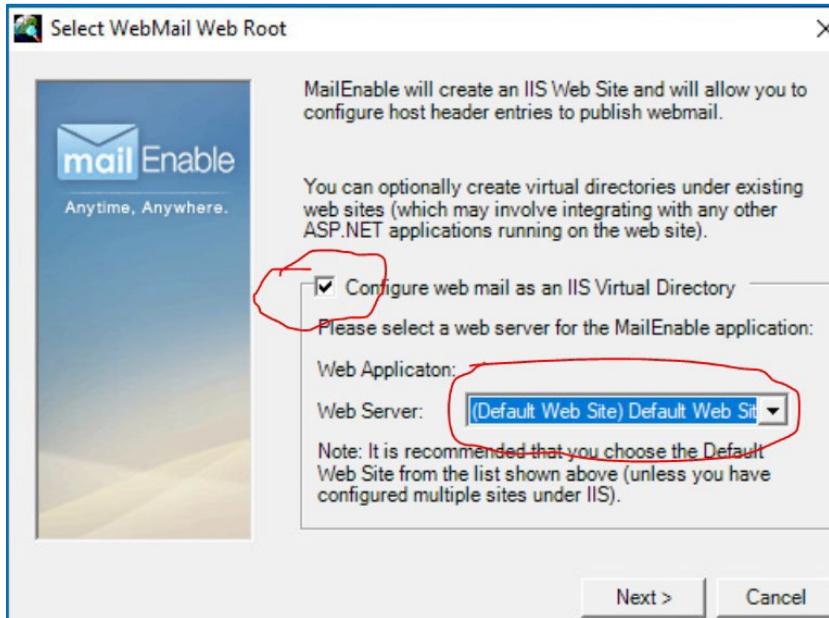
Figure 31 – Example SMTP Connector Configuration



- k. Hit next a couple more times, until you come to the "Select Webmail Web Root," windows. Make sure that the box is checked and the default web site is selected (it's recommended!). If not, that means you did not install the Web Server IIS role (shame on you ♻).

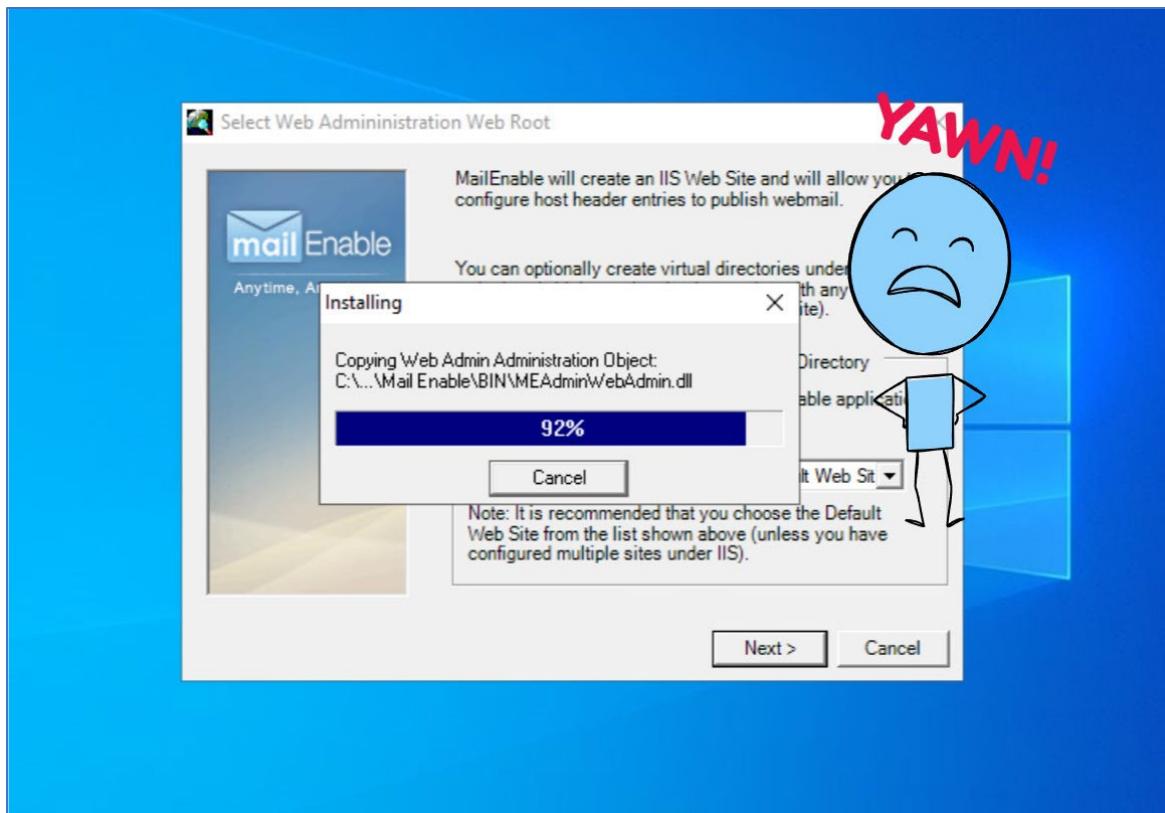
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 32 - Webmail Web Root



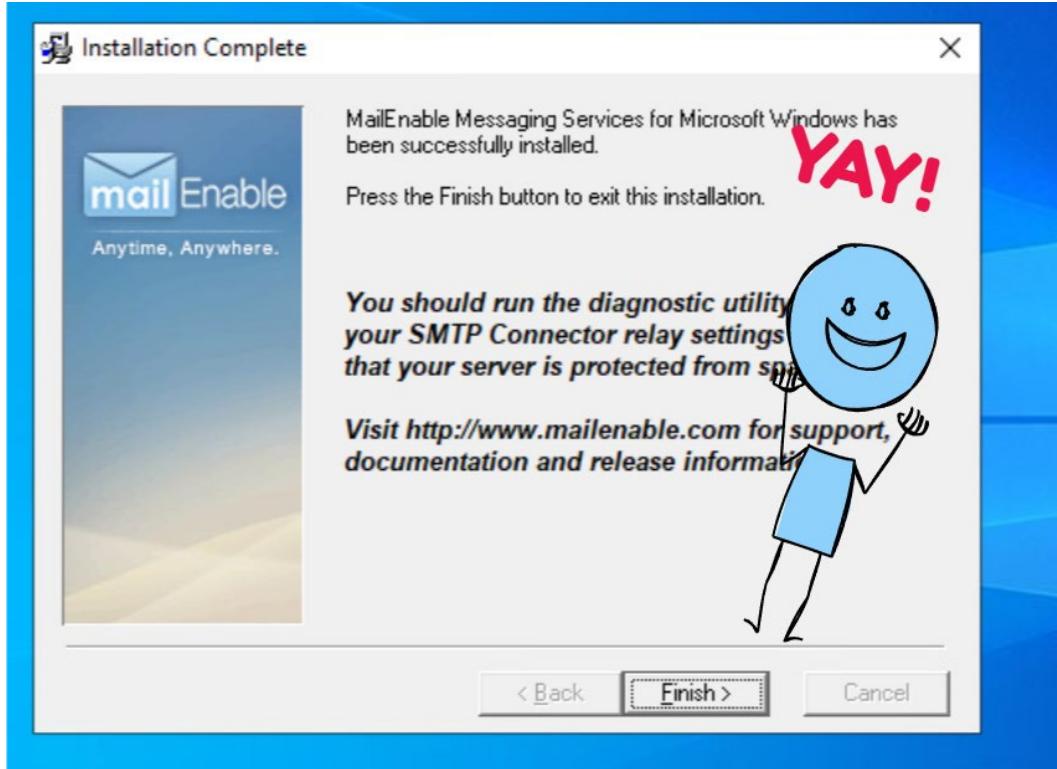
- I. Hit Next again and wait....BE PATIENT.... while the install completes.

Figure 33 – YAWN!



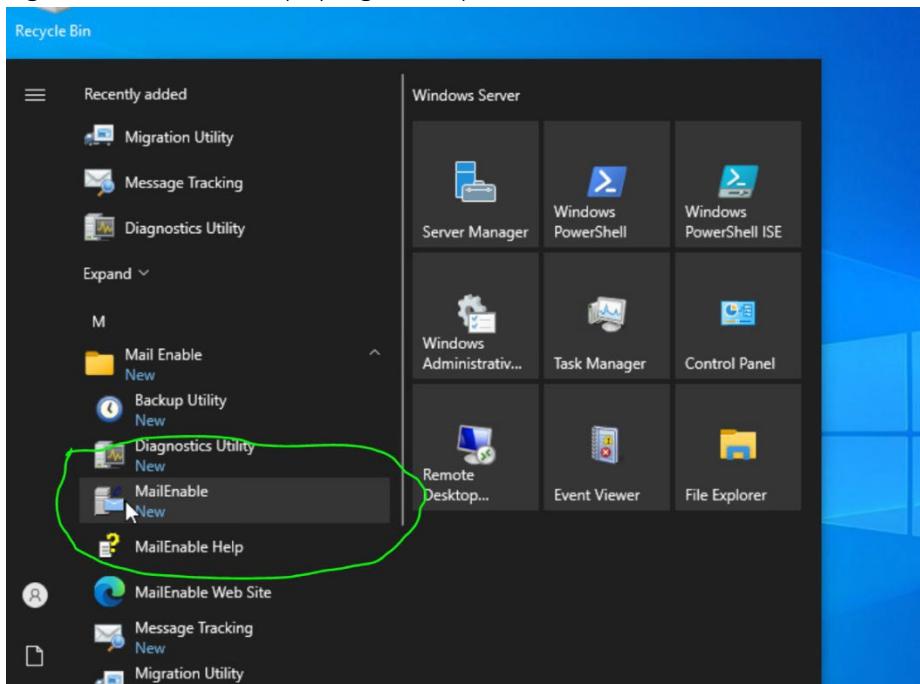
- j. Hit Next again and then Finish.

Figure 34 – Installation Complete



- k. When you done lots of stuff with your web browser is going to happen, just ignore it and close your browser.
- l. You can find MailEnable in your programs list (Figure 34), select it to open the program.

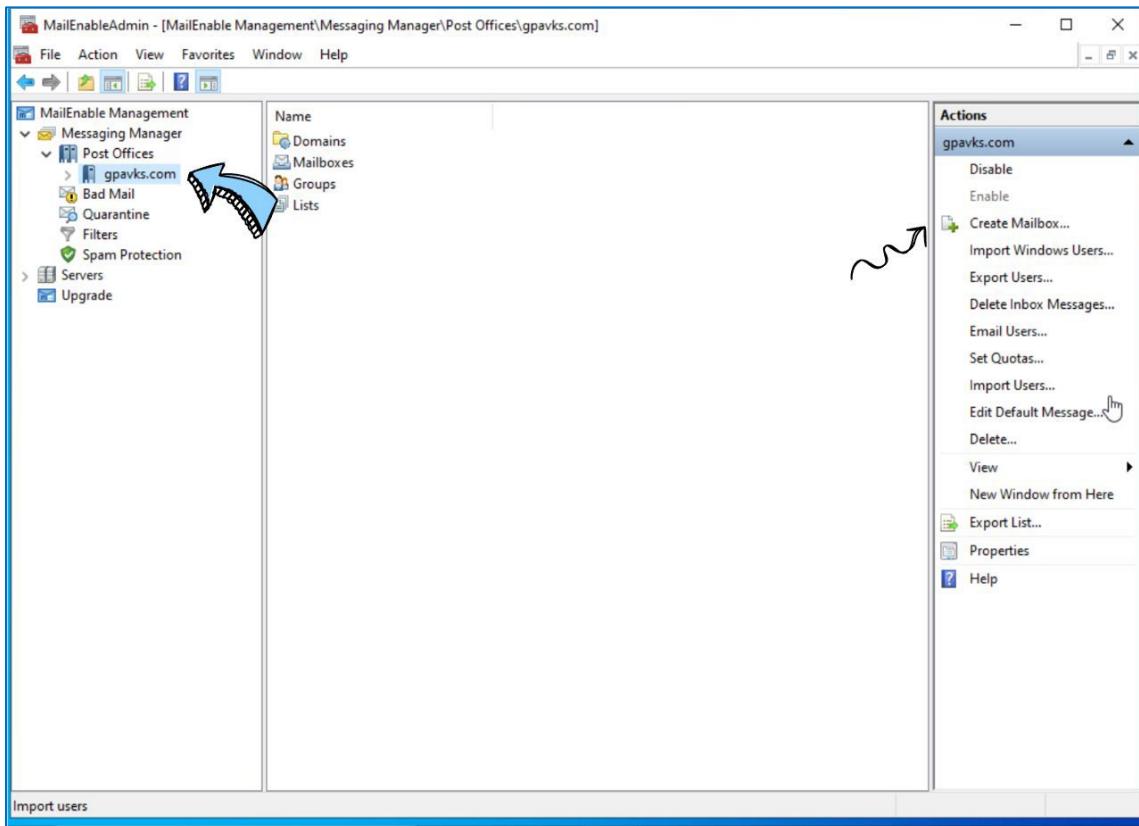
Figure 34 – MailEnable (in program list)



RIT | Golisano College of Computing and Information Sciences School of Information

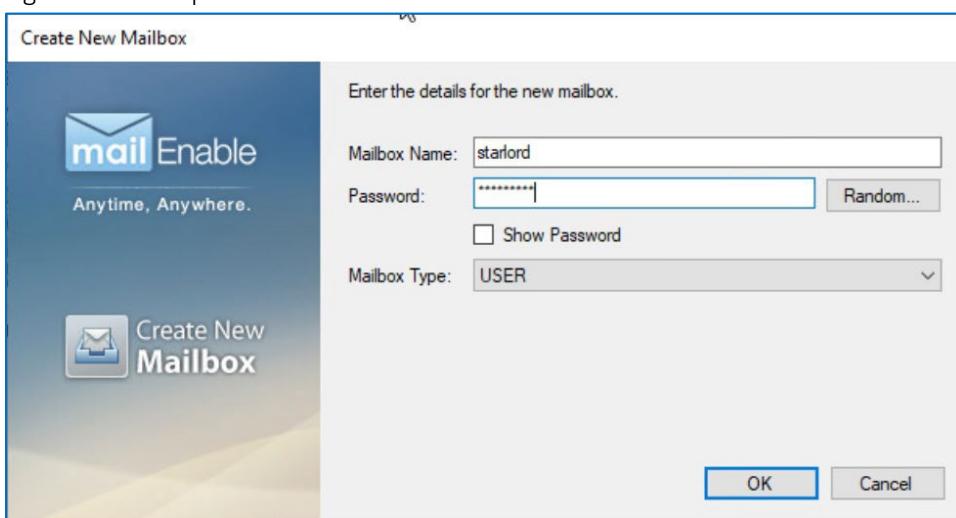
- m. Next, we need to create a couple mail boxes. To do this, from the MailEnable Management menu, expand “Messaging Manager,” then “Post Offices,” select your domain and on the right under *Actions*, select *Create Mailbox*.

Figure 35 – Create Mailbox



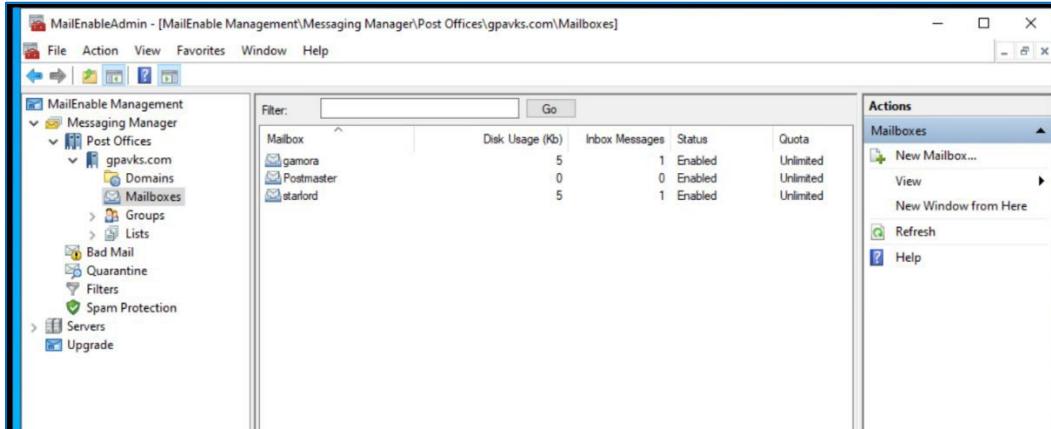
- n. These instructions use starlord and gamora as the mailboxes.

Figure 36 – Sample Mailbox Creation



- o. For reference, Figure 37 shows the mailboxes listed for the post office gpavks.com.

Figure 37 – Mailboxes for gpavks.com



The screenshot shows the MailEnable Management interface. The left sidebar navigation tree includes 'MailEnable Management', 'Messaging Manager' (selected), 'Post Offices' (selected), 'gpavks.com', 'Domains', 'Mailboxes' (selected), 'Groups', 'Lists', 'Bad Mail', 'Quarantine', 'Filters', 'Spam Protection', 'Servers', and 'Upgrade'. The main pane displays a table of mailboxes with columns: Mailbox, Disk Usage (Kb), Inbox Messages, Status, and Quota. The data is as follows:

Mailbox	Disk Usage (Kb)	Inbox Messages	Status	Quota
gamora	5	1	Enabled	Unlimited
Postmaster	0	0	Enabled	Unlimited
starford	5	1	Enabled	Unlimited

The right pane shows an 'Actions' menu with options: Mailboxes (selected), New Mailbox..., View, New Window from Here, Refresh, and Help.

Activity 6 – Email Client Configuration

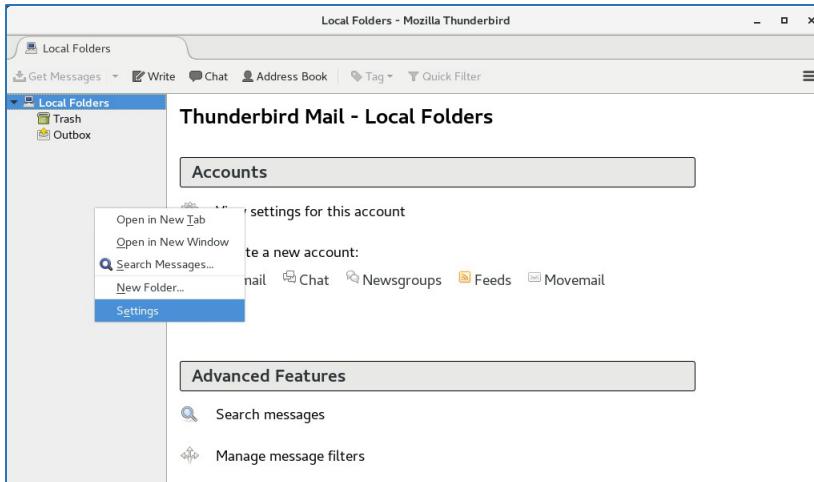
For this activity, you will configure Linux and Windows email clients. The screenshots and examples provided in this activity use the Thunderbird for Rocky Linux 8. For Windows 10,

- a. Install Thunderbird on Rocky Linux (At this point installing software on Linux should be familiar to you, if not, a quick Google search should help. Here's a little hint.

<https://www.thunderbird.net/en-US/>

- b. Launch Thunderbird. To add an account, right click on “Local Folders,” and select settings.

Figure 38 - Settings

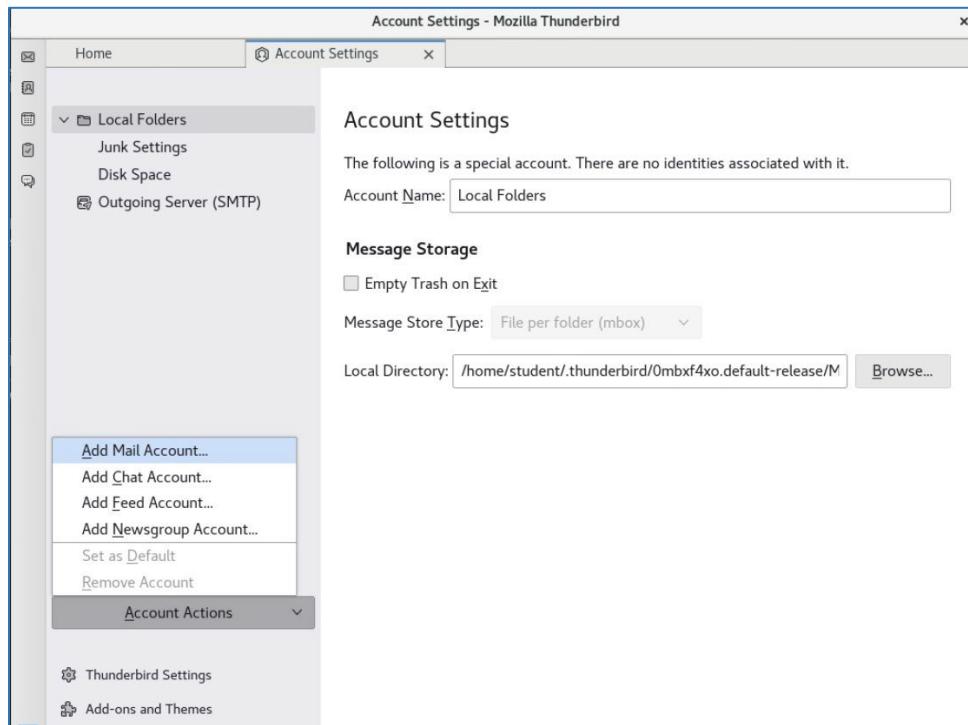


The screenshot shows the Mozilla Thunderbird 'Local Folders' window. The left sidebar shows 'Local Folders' with 'Trash' and 'Outbox' options. The main pane has a 'Thunderbird Mail - Local Folders' header. A context menu is open over the 'Local Folders' icon, with 'Settings' selected. The menu also includes 'Open in New Tab', 'Open in New Window', 'Search Messages...', 'New Folder...', and 'Account Actions...'. Below the menu, there are sections for 'Accounts' (with a sub-menu for 'settings for this account') and 'Advanced Features' (with 'Search messages' and 'Manage message filters').

- c. From the “Account Settings” window select “Account Actions” and the “Add Mail Account... ”.

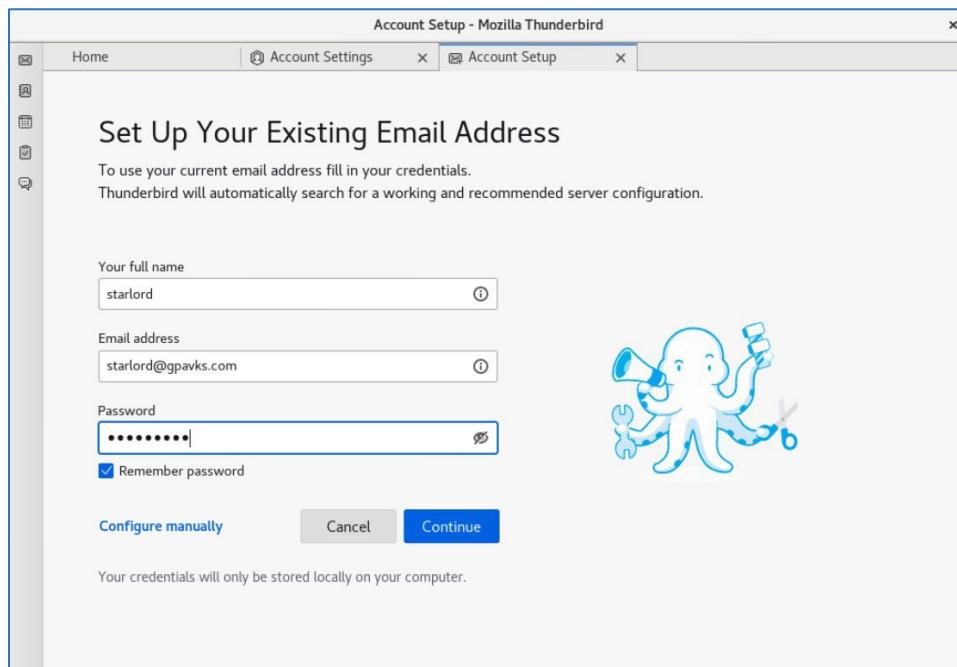
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 39 – Add Account



- d. Enter the user information for one of the accounts. (Figure 40). Click *Continue*.

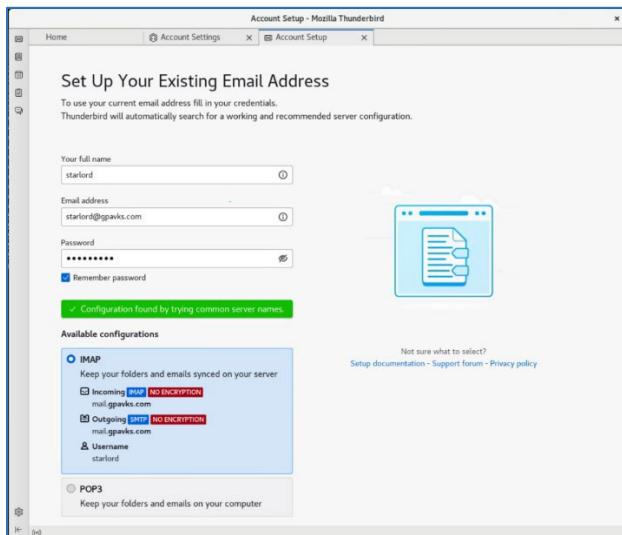
Figure 40 - Add User Account



- e. If you have DNS configured correctly (especially the MX resource record), then Thunderbird will find your mail server automatically and all you need to do is click Done.

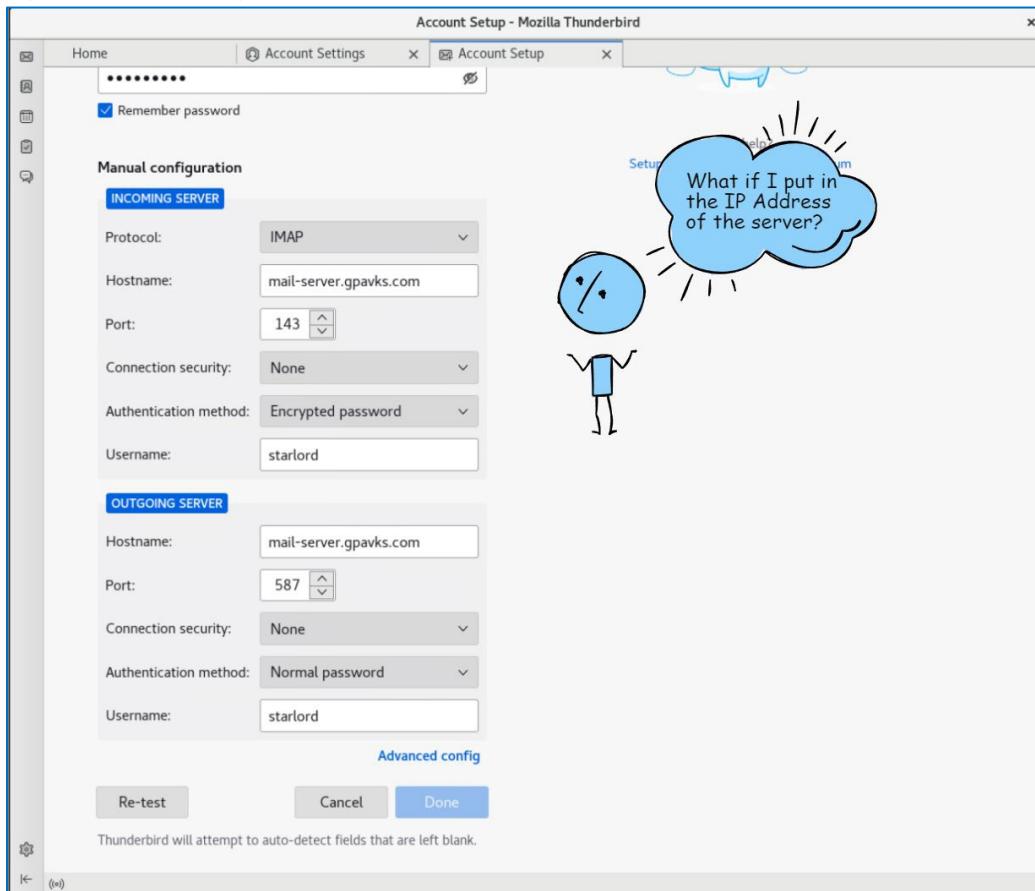
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 41 – Configuration Found



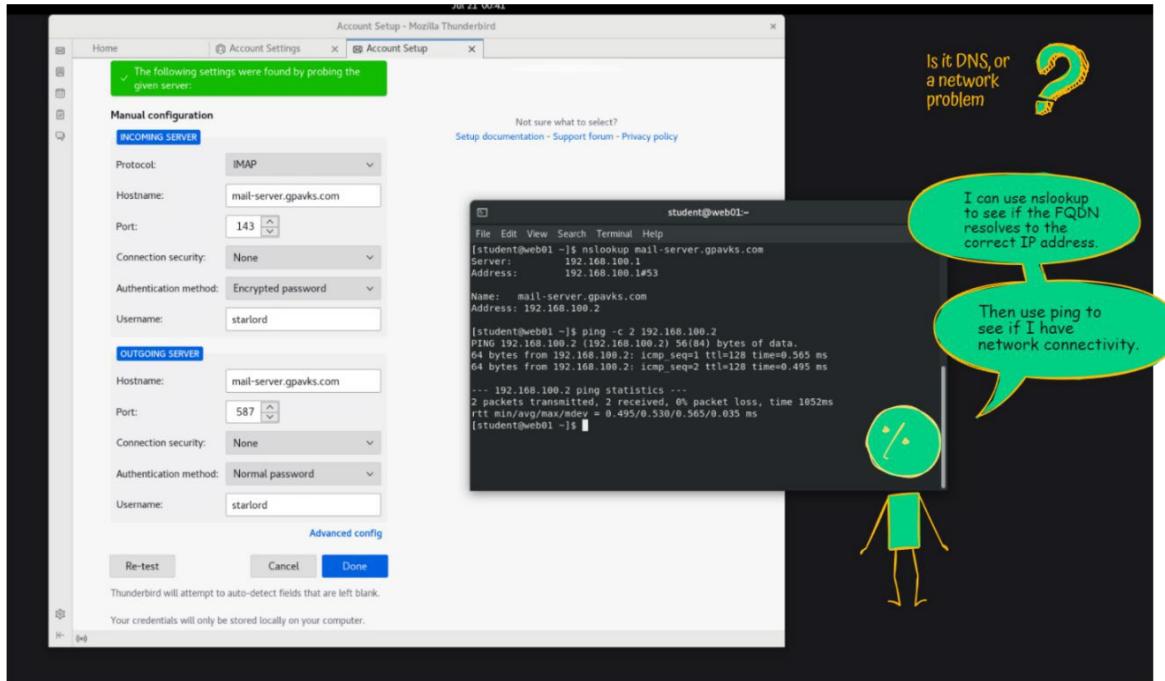
- f. If not, you'll need to do some troubleshooting. By default, Thunderbird uses the FQDN of the mail server. Select "Manual Configuration," and check to see if it is using the FQDN, if not enter it. If it is still not connecting try entering the IP Address of the server, if that works then your issue is with DNS. If not then check firewall settings, trying pinging the mail server from the client to make sure there is network connectivity. Does DNS resolve to the correct IP for the mail server (Figure 43)?

Figure 42 – Checking the Mail Servers FQDN



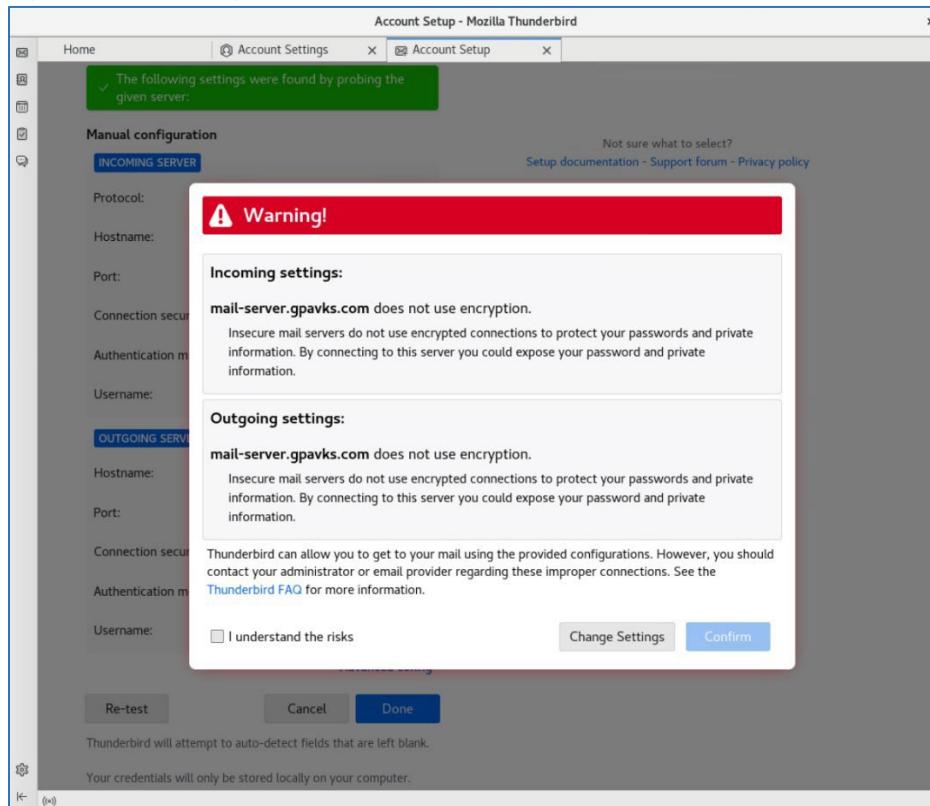
RIT | Golisano College of Computing and Information Sciences School of Information

Figure 43 – Troubleshooting with nslookup and ping



- Once the correct hostname for the mail server is configured and the ports are populated, click **Done**. Because we did not configure Transport Layer Security for the mail server, a warning is shown (Figure 44), check the box “I understand the risks.” Click “Confirm,” and then “Finish”.

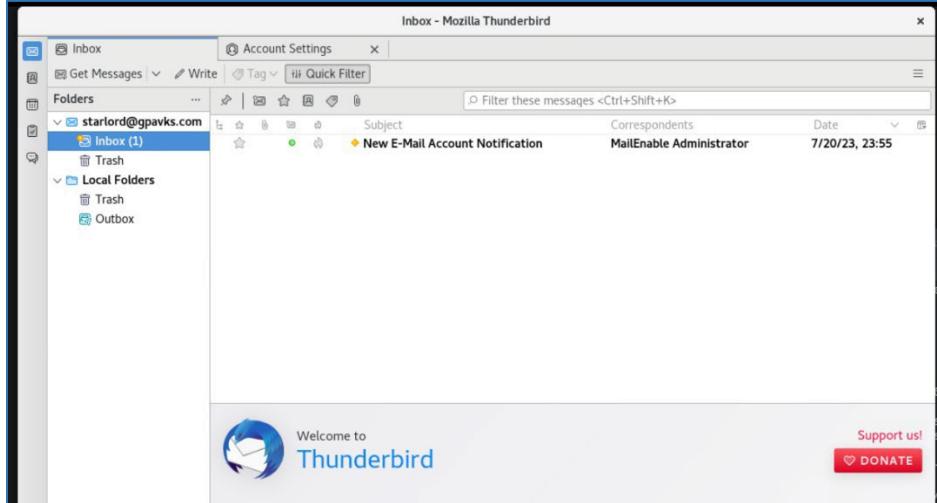
Figure 44 – Warning



RIT | Golisano College of Computing and Information Sciences School of Information

- h. If everything goes well, you should see the mailbox and an email notification (Figure 45). To do a quick test, try sending an email to the user to see if they receive it. Using this as an example, starlord would send an email to itself.

Figure 45 – Account Notification



- i. Now it's time for you to take charge. Install an MUA on Windows 10 and configure it. There are several free and open-source mail clients available. You can use Thunderbird, or give another one a try, like eM.



For the report, include two screenshots. The first screenshot will show the email being sent from one client and received by the other. In these examples, the email is being sent from the Windows client by Gamora (Figure 46), and received on the Linux client by Starlord (Figure 47).

Figure 46 – Sent Email from Windows Client

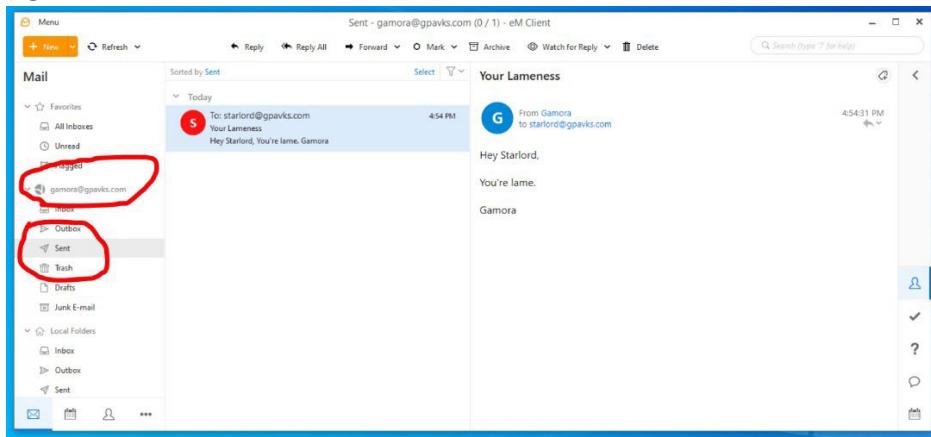
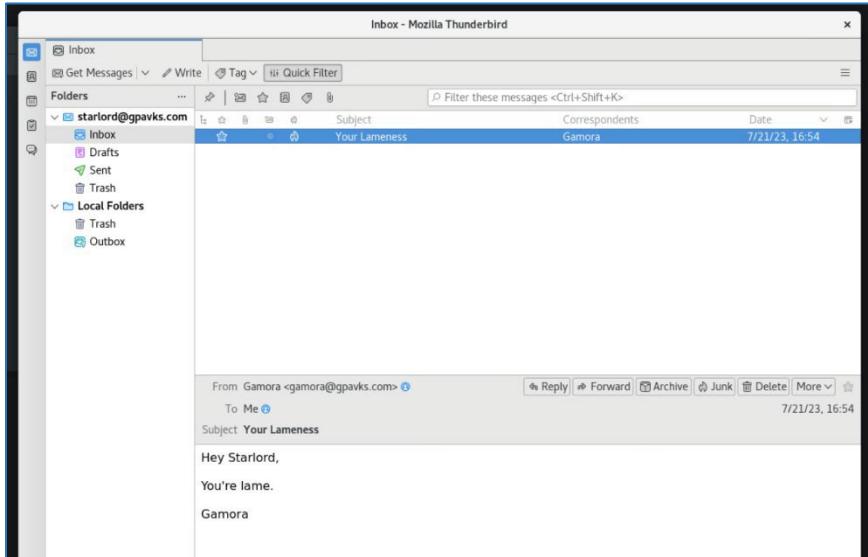


Figure 47 – Received Email on Linux Client



Activity 7 – Capturing Email Network Traffic

For this activity, you will be using Wireshark to capture IMAP and SMTP traffic.

- Launch Wireshark on the email server.
- Send an email from one account to another, it does not matter who the sender or the receiver is for this exercise.
- Capture the traffic and filter for IMAP and SMTP traffic (Figures 48 and 49).

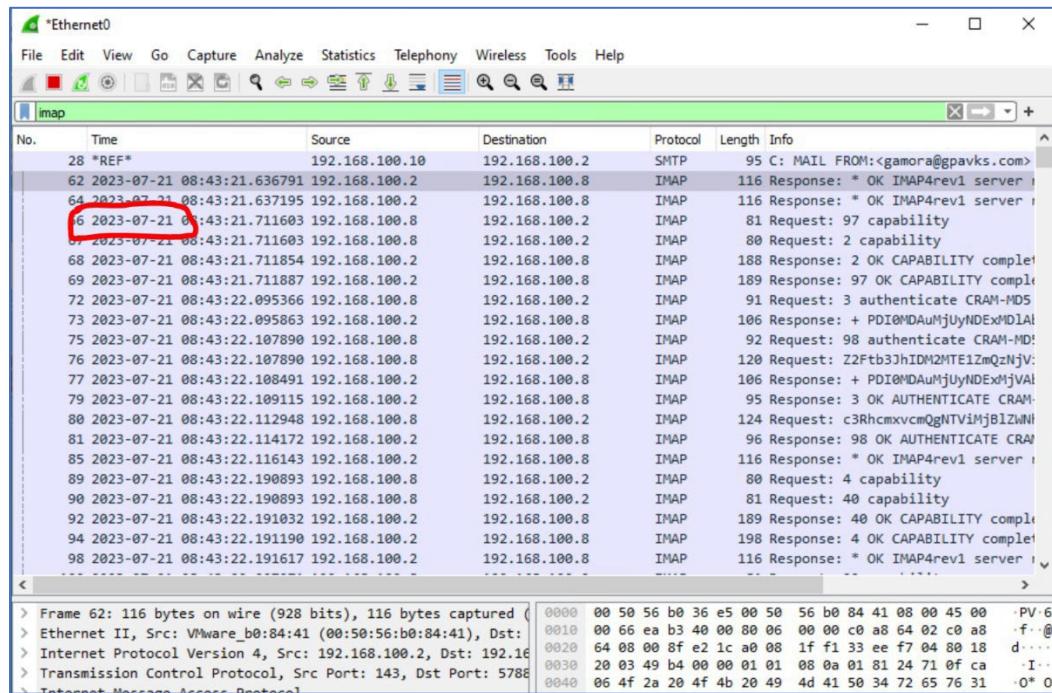
Figure 48 – Example SMTP Network Trace

The screenshot shows the Wireshark interface with the title bar "Ethernet0". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar has various icons for file operations and analysis. A search bar at the top right contains the text "smtp". The main window displays a list of captured network frames. The first few frames are highlighted with a red oval, showing the initial SMTP handshake between the client and the server. The table below provides a detailed view of the captured data:

No.	Timestamp	Source	Destination	Protocol	Length	Info
15	2023-07-21 08:41:05.927291	192.168.100.2	192.168.100.10	SMTP	152	S: 220 MAIL-SERVER.gpavks.com ESMTP
20	2023-07-21 08:41:05.927738	192.168.100.10	192.168.100.2	SMTP	77	C: EHLO [192.168.100.10]
21	2023-07-21 08:41:05.927995	192.168.100.2	192.168.100.10	SMTP	181	S: 250-gpavks.com [192.168.100.10]
22	2023-07-21 08:41:05.928472	192.168.100.10	192.168.100.2	SMTP	66	C: AUTH LOGIN
23	2023-07-21 08:41:05.928701	192.168.100.2	192.168.100.10	SMTP	72	S: 334 VXNlcm5hbWU6
24	2023-07-21 08:41:05.928877	192.168.100.10	192.168.100.2	SMTP	80	C: User: Z2Ftb3JhQGdwYXZrcy5jb20-
25	2023-07-21 08:41:05.929113	192.168.100.2	192.168.100.10	SMTP	72	S: 334 UGFzc3dvcnQ6
26	2023-07-21 08:41:05.929437	192.168.100.10	192.168.100.2	SMTP	68	C: Pass: UGFzc3dvcnQx
27	2023-07-21 08:41:05.930367	192.168.100.2	192.168.100.10	SMTP	73	S: 235 Authenticated
28	*REF*	192.168.100.10	192.168.100.2	SMTP	95	C: MAIL FROM:<gamora@gpavks.com>
29	2023-07-21 08:41:05.930802	192.168.100.2	192.168.100.10	SMTP	97	S: 250 Requested mail action okay
30	2023-07-21 08:41:05.930979	192.168.100.10	192.168.100.2	SMTP	85	C: RCPT TO:<starlord@gpavks.com>
31	2023-07-21 08:41:05.932000	192.168.100.2	192.168.100.10	SMTP	97	S: 250 Requested mail action okay
32	2023-07-21 08:41:05.932186	192.168.100.10	192.168.100.2	SMTP	60	C: DATA
33	2023-07-21 08:41:05.941642	192.168.100.2	192.168.100.10	SMTP	100	S: 354 Start mail input; end with .
36	2023-07-21 08:41:05.962053	192.168.100.10	192.168.100.2	SMTP	68	[TCP Previous segment not captured]
43	2023-07-21 08:41:06.275316	192.168.100.2	192.168.100.10	SMTP	97	S: 250 Requested mail action okay
44	2023-07-21 08:41:06.275758	192.168.100.10	192.168.100.2	SMTP	60	C: QUIT
45	2023-07-21 08:41:06.275887	192.168.100.2	192.168.100.10	SMTP	96	S: 221 Service closing transmission channel
16101	2023-07-21 14:13:03.767397	192.168.100.2	192.168.100.10	SMTP	152	S: 220 MAIL-SERVER.gpavks.com ESMTP
16102	2023-07-21 14:13:03.767865	192.168.100.10	192.168.100.2	SMTP	77	C: EHLO [192.168.100.10]

At the bottom, there is a status bar with the message "Frame 28: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)" and a hex dump of the captured data.

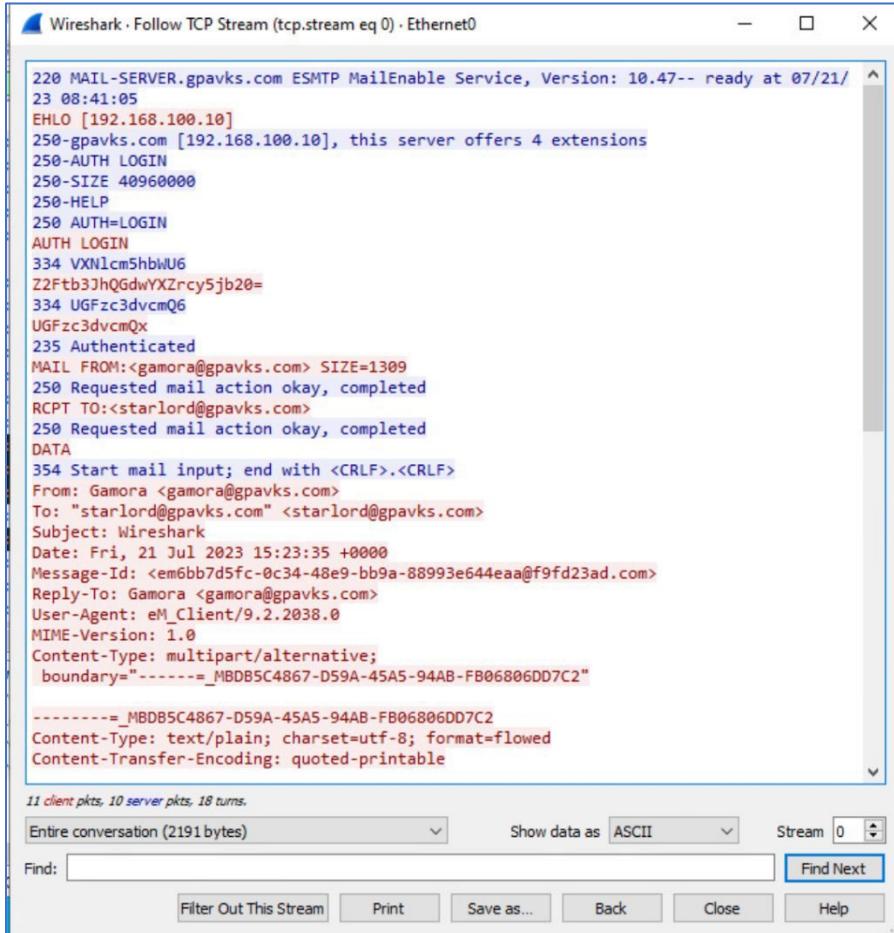
Figure 49 – Example IMAP Network Trace



For the report, include screenshots from the email network traces. The traces must show the IP address of the server and clients involved in the communication and the date/time stamp must be visible. Use the screenshots to explain the communication that occurs between the client and server when forwarding and receiving the email message. The explanation must be included with the screenshot, in the report. If you do not explain the communication, you will not receive credit. The explanation must be in your own words.

You may also follow the TCP stream (Figure 65) and include that screenshot, in addition to the other traces for your explanation.

Figure 50 – Example Screenshot of “Following the TCP Stream”



Activity 8 – Remote Logging

Rsyslog serves the purpose of logging and organizing system messages according to specific categories for storage. For this task, you have the option to either add another Rocky Linux 8 virtual machine to your existing infrastructure (requiring two virtual machines) or use the ones you already have. One of these devices will act as the centralized log server responsible for collecting logs, while the other will be responsible for sending the logs to the central server. The ability to aggregate logs on a central server is vital for most organizations as it greatly simplifies log management. Moreover, it aligns with NIST recommendations and plays a crucial role in enhancing security, making it an indispensable component for establishing a robust log management system in an enterprise environment.

- Select a Rocky Linux VM to be the centralized log server. Again, you can deploy another Rocky VM, or use an existing one.
- Update the system and make any necessary configuration changes.
- You may disable SELinux or set it to permissive mode.
- By default, rsyslog is installed on Rocky Linux 8; to see the version type `rpm -qa rsyslog` into the terminal.
- Next, edit the rsyslog configuration file in `/etc/rsyslog.conf`. Rsyslog can be configured to receive logs via TCP or UDP. For this activity we'll configure both.
- Uncomment the following lines in the file by removing the # symbol.

```
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")
```

```
module(load="imtcp") # needs to be done just once
input(type="imtcp" port="514")
```

Figure 51 – Rsyslog Configuration Edit

```
student@web01:~$ gr te
File Edit View Search Terminal Help
# or latest version online at http://www.rsyslog.com/doc/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

#### MODULES ####

module(load="imuxsock"      # provides support for local system logging (e.g. via
logger command)
      SysSock.Use="off") # Turn off message reception via local log socket;
                          # local messages are retrieved through imjournal now.
module(load="imjournal"          # provides access to the systemd journal
      StateFile="imjournal.state") # File to store the position in the journal
#module(load="imklog") # reads kernel messages (the same are read from journald)
#module(load="immark") # provides --MARK-- message capability

# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")

# Provides TCP syslog reception
# for parameters see http://www.rsyslog.com/doc/imtcp.html
module(load="imtcp") # needs to be done just once
input(type="imtcp" port="514")

#### GLOBAL DIRECTIVES ####

# Where to place auxiliary files
global(workDirectory="/var/lib/rsyslog")

# Use default timestamp format
```

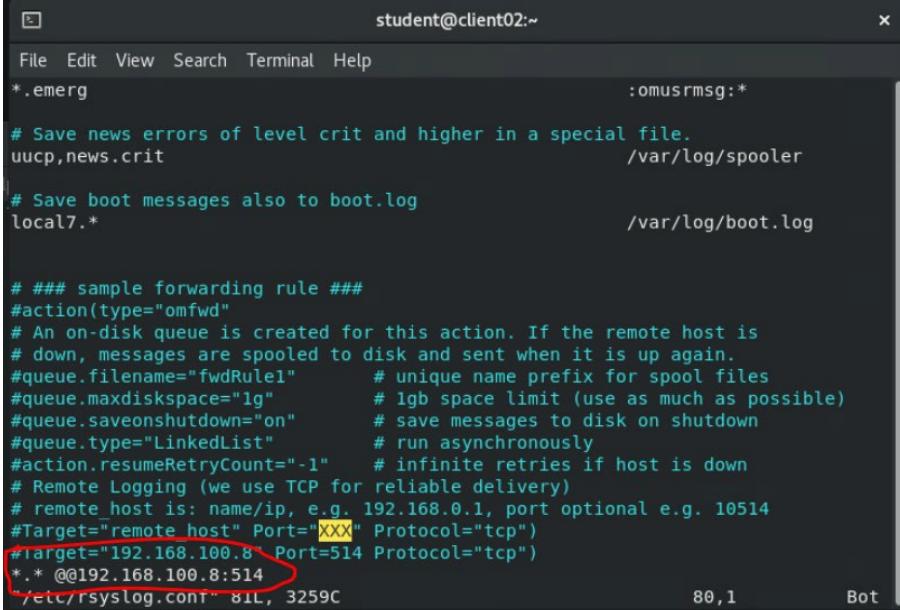
- g. Next, create a firewall rule to allow traffic through the port rsyslog is listening on, which is port 514. In addition, allow TCP and UDP traffic. Yes, enable the firewall.

```
$sudo firewall-cmd --permanent --add-port=514/tcp
$sudo firewall-cmd --permanent --add-port=514/udp
$sudo firewall-cmd --reload
```

- h. Restart the rsyslog and firewall services.
- i. Next, configure the client to forward logs to the centralized log server. To do this, you will need to edit the rsyslog configuration file on the client. In the configuration file add the following line to the end of the file.

```
*.* @@<server IP Address or hostname>:514
```

j. Figure 52 – Client rsyslog Configuration File



```

student@client02:~$ cat /etc/rsyslog.conf
*.emerg                                     :omusrmmsg:*
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                /var/log/spooler

# Save boot messages also to boot.log
local7.*                                      /var/log/boot.log

# ### sample forwarding rule #####
#action(type="omfwd"
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
#queue.filename="fwdRule1"                      # unique name prefix for spool files
#queue.maxdiskspace="1g"                        # 1gb space limit (use as much as possible)
#queue.saveonshutdown="on"                      # save messages to disk on shutdown
#queue.type="LinkedList"                       # run asynchronously
#action.resumeRetryCount="-1"                   # infinite retries if host is down
# Remote Logging (we use TCP for reliable delivery)
# remote_host is: name/ip, e.g. 192.168.0.1, port optional e.g. 10514
#Target="remote host" Port="XXX" Protocol="tcp"
#target="192.168.100.8" Port=514 Protocol="tcp"
*.* @192.168.100.8:514
/etc/rsyslog.conf" 81L 3259C

```

k. Restart rsyslog on the client.

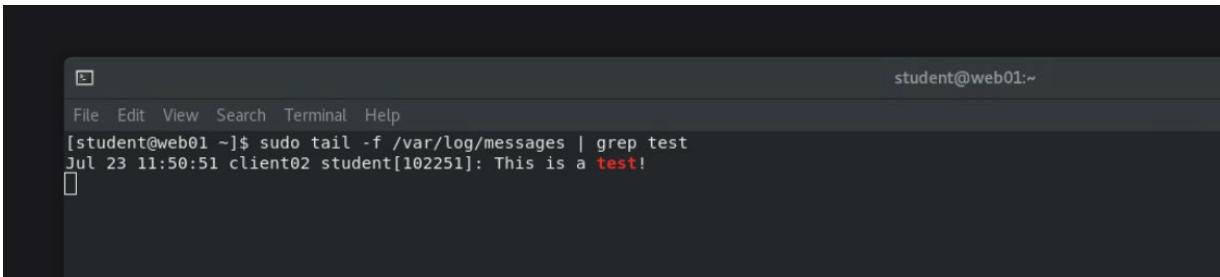
l. Use the following command on the central log server to capture messages.

```
$ tail -f var/log/messages
```

m. Using the **logger** command, test to verify that the central log server is receiving messages from the webserver.

```
$ logger This is a test!
```

Figure 53 – Sample Log



```

student@web01:~$ sudo tail -f /var/log/messages | grep test
Jul 23 11:50:51 client02 student[102251]: This is a test!

```

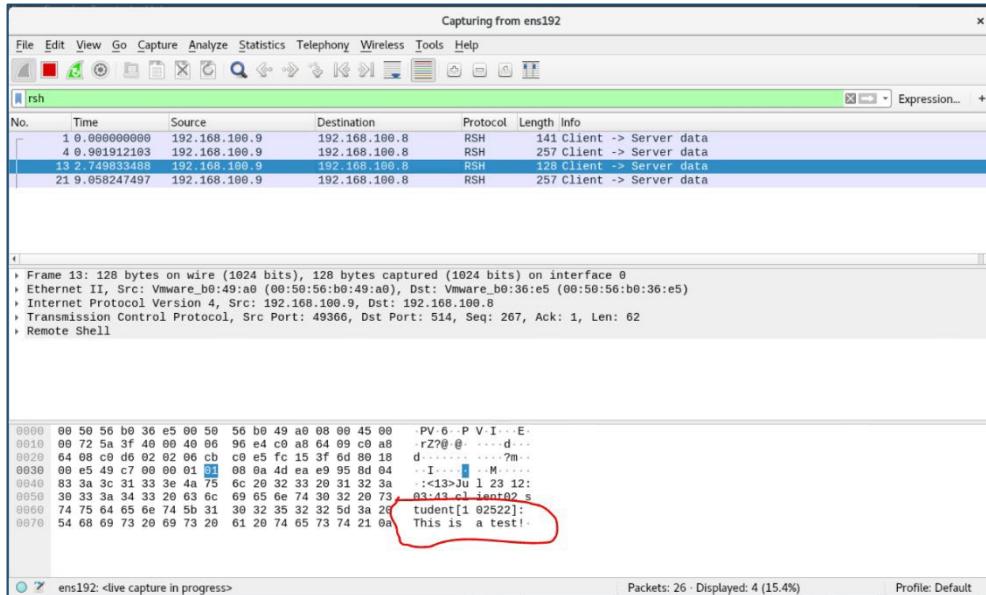


For the report, include a single screenshot showing that the rsyslog server received the message. Refer to Figure 53 for an example. The image must show the hostname of the client and the message.

RIT | Golisano College of Computing and Information Sciences School of Information

Additionally, capture a Wireshark trace of the exchange, for brevity, you can filter using "rsh". The trace must include the message and the IP addresses of the client and log server. Refer to Figure 54 for an example.

Figure 54 – Network Trace Example



Activity 9 – The Cron Service

Crond is a daemon process designed for automating tasks, allowing them to run in the background without user intervention. A "cron job" refers to a scheduled task that executes at specific intervals. To manage these tasks, the main file used is called the crontab, which instructs the cron daemon on when to run each task. In this context, Figure 55 represents an example of a crontab file typically located in the /etc directory.

Before you proceed with configuring cron jobs, it's essential to familiarize yourself with the file's format. Each user has their crontab file, which can be found in the /var/spool/cron directory. However, please note that the file for a user is created only after they run their first cron job.

Figure 55 – Crontab file format

```
[root@webserver etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# ----- minute (0 - 59)
# | ----- hour (0 - 23)
# | | ----- day of month (1 - 31)
# | | | ----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | ----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command to be executed

[root@webserver etc]#
```

Example 1:

```
30 08 10 06 * /home/backups/full-backup.sh
```

The first example runs the shell script “*full-backup.sh*”, every June (06), on the tenth day (10) at 8:30am. Note that the time value is in the 24-hour format, if it was 8:30 PM, the hour would be 20 and not 8. The use of the asterisk means to run this job every day of the week.

Example 2:

```
00 11, 16 * * * /home/bin/incremental-backup.py
```

This example introduces something a little more interesting. The following is a Python script. However, notice the “11, 16” in the hour column, this means to run the job during the 0th minute (00) at 11 AM (11) and 4 PM (16), every day, every month, and every day of the week.

Now that we have an idea of what information goes into the crontab file how do we go about adding or editing entries in the file. To add an entry to the crontab file use the **crontab -e** command as the current user. The user that creates the job must also be the same user that is the owner of the script.

Please Note: Red Hat recommends that you do not edit /etc/crontab directly! Use the crontab utility!

The following commands provide a summary of the crontab command and some of its options.

crontab -l	lists the user's crontab.
crontab -e	edits the user's crontab.
crontab -r	deletes the user's crontab.

Of course, reviewing the crontab man page will help too.

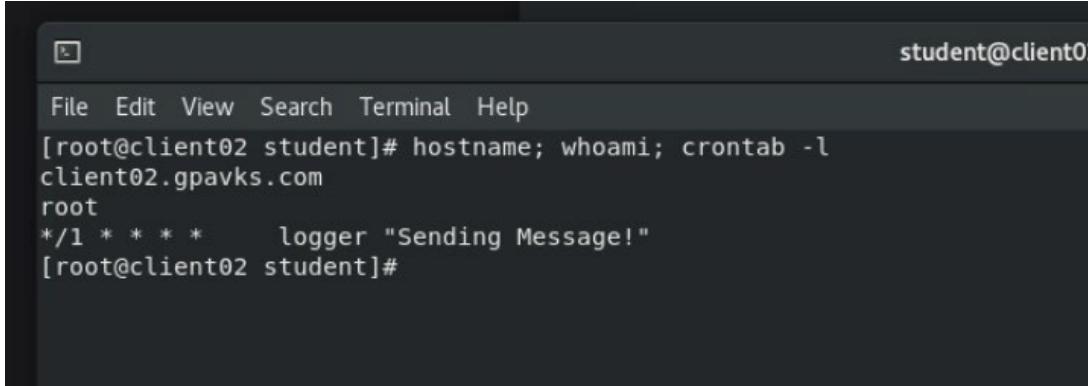
Create a cron job that uses the **logger** command to send a message from the client to the central log server every minute.



For the report, include a screenshot showing the output of the **hostname**, **whoami**, and **crontab -l** commands. See Figure 56 for an example.

RIT | Golisano College of Computing and Information Sciences School of Information

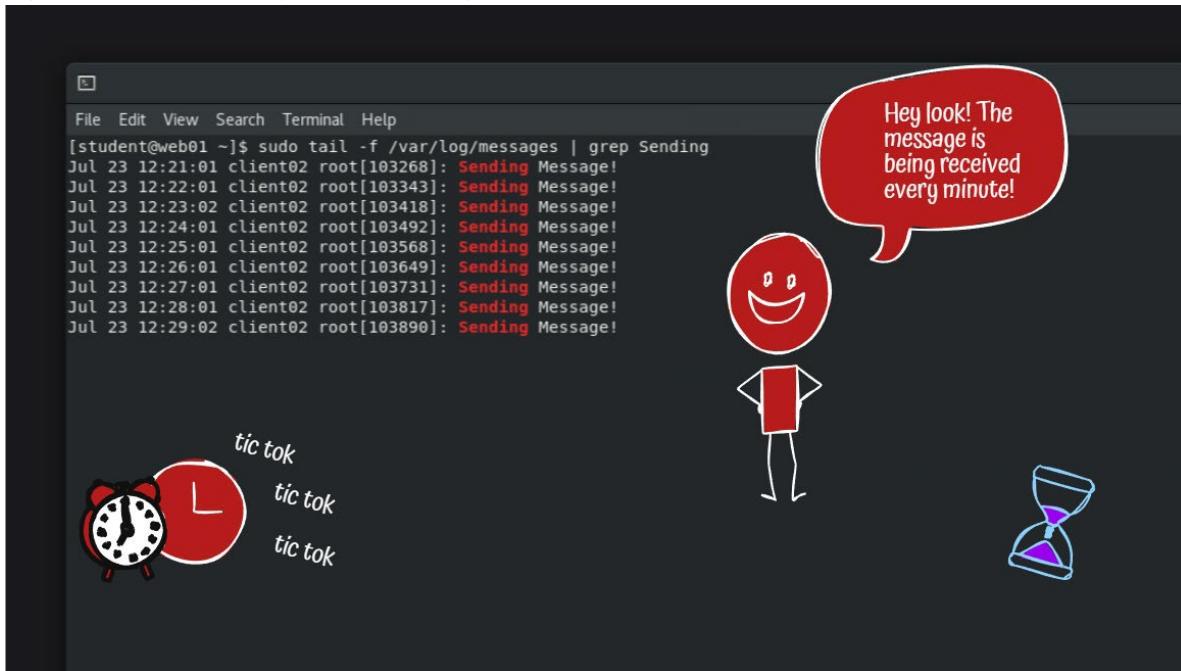
Figure 56 – Example Cronjob List Validation



A screenshot of a terminal window titled "student@client02". The window shows the command "hostname; whoami; crontab -l" being run. The output indicates the host is "client02.gpavks.com" and the user is "root". A cron job is listed: "* * * * * logger \"Sending Message!\"". The prompt "[root@client02 student]#" is visible at the end.

Using the **tail -f /var/log/messages** command, provide a second screenshot showing that the log server received the message. The screenshot must show at least two instances of the message received with the time stamp showing a 2-minute interval. It must be the same message that appears in the cronjob list. Refer to Figure 72 for an example.

Figure 57 – Sample Output from the Log Server



A screenshot of a terminal window titled "[student@web01 ~]\$". The window shows the command "sudo tail -f /var/log/messages | grep Sending" being run. The output displays multiple entries of the message "Sending Message!" at various times on July 23, with a 2-minute interval between consecutive entries. To the right of the terminal, there is a cartoon illustration of a red stick figure with a smiling face, a speech bubble saying "Hey look! The message is being received every minute!", a red alarm clock with the text "tic tok" next to it, and a blue hourglass.

All screenshots for Lab 06 must be included in the lab report. For each missing screenshot, you will receive a 5% penalty on the report grade. If your screenshots do not include the required information, are illegible, blurry, or otherwise unreadable, you will not receive credit. Any attempt to alter the information in the screenshots in any way is academic dishonesty, and you will receive a zero grade for the report.

SCREENSHOT SUMMARY

Figure 1 – Apache Server Verification

Figure 2 – DNS Resource Record Configuration

Figure 3 – Default Site Verification

Figure 4 – First Virtual Host Site

Figure 5 – Second Virtual Host Site

Figure 6 – Certificate Verification

Figure 7 – Secure Site

Figure 8 – SSL Virtual Host File

Figure 8 – Sent Email Verification

Figure 9 – Received Email Verification

Figure 10 – SMTP Network Trace

Figure 11 – IMAP Network Trace

Figure 12 – Rsyslog Message Verification

Figure 13 – Cronjob List Validation

Figure 14 – Cronjob Message Verification