

Lab 10 Requirements

- Ubuntu Web Server VM Snapshot from Lab 09

Part 01: Install Clam Anti-Virus

We will install the Clam Anti-Virus software on the Ubuntu Web server

```
apt-get update
apt-get install clamav clamav-daemon clamav-freshclam
```

You can manually update the AV feed by entering the following command: `freshclam`

But because you are running a background process that does this for you, you will receive an error:

```
root@artmack-uws:/home/artmack# freshclam
ERROR: /var/log/clamav/freshclam.log is locked by another process
```

The Clam Anti-Virus should already be running... To ensure that it starts up on boot, add it to the services that automatically start up

```
update-rc.d clamav-daemon defaults
update-rc.d clamav-freshclam defaults
service clamav-daemon start
```

Part 02: Create a script to scan the web server

As discussed, any commands issued in the terminal can be put into a script. Scripts can then be automated so that the commands run as scheduled. Create a new file called **simple_scan.sh** and enter the following into the file:

```
#!/bin/bash
SCAN_DIR="/home"
LOG_FILE="/var/log/clamav/manual_clamscan.log"

/usr/bin/clamscan -ri $SCAN_DIR >> $LOG_FILE
```

Save the file and ensure that the script has execute permissions

Run the script

```
root@artmack-uws:/home/artmack# chmod 755 simple_scan.sh
root@artmack-uws:/home/artmack# ./simple_scan.sh
root@artmack-uws:/home/artmack# _
```

There will be no output when the script has finished executing. You can see the results in the log file you specified in your script...

Slide 01:

- Take a screenshot showing the contents of the log file and place it into slide 01
- Include your **FOLusername**

Part 03: Install and configure Send Mail

Install Send Mail and related utilities

```
apt-get install sendmail mailutils sendmail-bin
```

The lab example will be using a gmail account in order to relay emails from the Ubuntu Web Server. To accomplish this, you will need to do two things:

First ensure that your gmail account will be able to authenticate your web server.

Go to Gmail, and select **Manage Google Account -> Security -> 2-Step Verification**

Scroll down to where you see **App Passwords**

← App passwords

Under **Select app**, choose **Mail**, then select a device

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

Chooser **Other** and give it a name

Click on **GENERATE**

You don't have any app passwords.

Select the app and device you want to generate the app password for.

ubuntu web server ×

GENERATE

You will see that Gmail has created an app password that can be used for your emails. This password will not be shown again, so ensure that you have copied it

Second, you will need to create a Gmail Authentication file:

```
mkdir -m 700 /etc/mail/authinfo/  
cd /etc/mail/authinfo/
```

Once in the new directory, create an auth file with a following content. (This file can have any name, in this example the name is **gmail-auth**):

```
AuthInfo: "U:root" "I:YOUR GMAIL EMAIL ADDRESS" "P:YOUR GENERATED APP PASSWORD"
```

**** The password you are using above is the one you generated for App passwords in Gmail ****

Create a hash map for the authentication file:

```
makemap hash gmail-auth < gmail-auth
```

Use a text editor such as **nano** to edit the send mail configuration file

Place the following lines into your **/etc/mail/sendmail.mc** right above the first "MAILER" definition line:

```
define(`SMART_HOST', `[smtp.gmail.com]')dnl  
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl  
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl  
define(`confAUTH_OPTIONS', `A p')dnl  
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl  
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl  
FEATURE(`authinfo', `hash -o /etc/mail/authinfo/gmail-auth.db')dnl
```

Once you have finished, save the changes to the file and exit

In the next step we will need to re-build sendmail's configuration. In order to do that, execute:

```
make -C /etc/mail
```

Reload sendmail service:

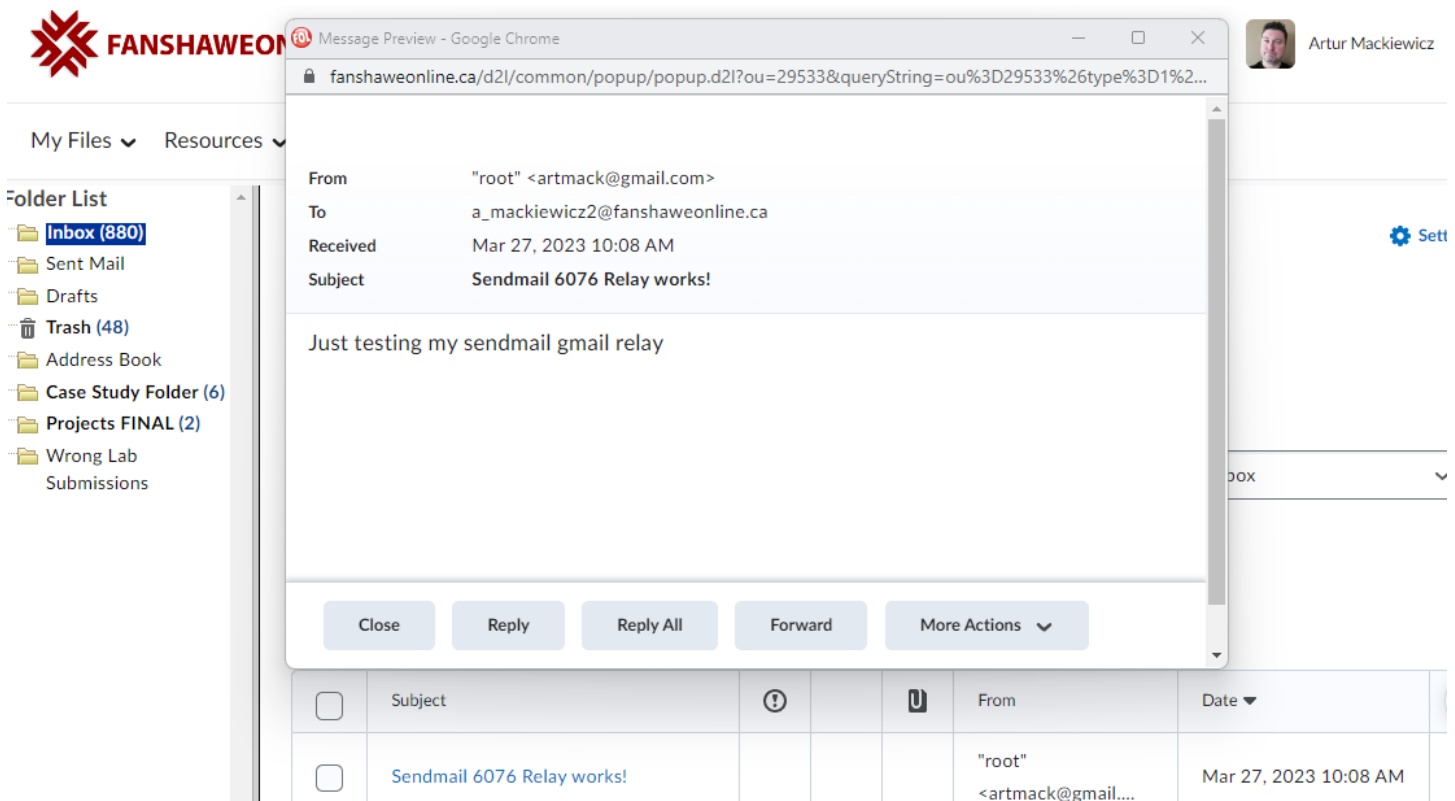
```
/etc/init.d/sendmail reload
```

Send test email from terminal using the following command:

```
echo "Just testing my sendmail gmail relay" | mail -s "Sendmail 6076 Relay works!" FOLusername@fanshaweonline.ca
```

Note: You need to replace FOLusername@fanshaweonline.ca with your FOL email address...

The email may take a minute or so to come in so be patient...



The screenshot shows a webmail interface with a sidebar on the left containing a 'Folder List' with items like 'Inbox (880)', 'Sent Mail', 'Drafts', 'Trash (48)', 'Address Book', 'Case Study Folder (6)', 'Projects FINAL (2)', 'Wrong Lab', and 'Submissions'. The main area displays a 'Message Preview' window for an email from 'root' <artmack@gmail.com> to 'a_mackiewicz2@fanshaweonline.ca'. The email subject is 'Sendmail 6076 Relay works!' and the body contains the text 'Just testing my sendmail gmail relay'. Below the preview window, a table lists the email in the inbox.

<input type="checkbox"/>	Subject	From	Date
<input type="checkbox"/>	Sendmail 6076 Relay works!	"root" <artmack@gmail....>	Mar 27, 2023 10:08 AM

Slide 02:

- Take a screenshot showing the email in your inbox as shown above and place it into slide 02

Part 04: Automate the AV scan and email alert with a cron job

In order to have the script run regularly, you will need to create a script and add it to the cron jobs

Create a new file called **auto_clam_scan** and place it in **/etc/cron.hourly/**

Enter the following code into the new file:

```
#!/bin/bash

# Email alert cron job script for ClamAV
# Email notifications in case of infections found will be sent

# Directories to scan
SCAN_DIR="/home /tmp /var"

# Location of log file
LOG_FILE="/var/log/clamav/auto_clam_scan.log"

# Uncomment to have scan remove files
#AGGRESSIVE=1
# Uncomment to have scan not remove files
AGGRESSIVE=0

# Email Subject
SUBJECT="Potential Threat Detected"
# Email To
EMAIL="FOLusername@fanshaweonline.ca"
# Email From
EMAIL_FROM="YOUR_TEST_GMAIL_ACCOUNT"

check_scan () {
    # If there were infected files detected, send email alert

    if [ `tail -n 12 ${LOG_FILE} | grep Infected | grep -v 0 | wc -l` != 0 ]
    then
        # Count number of infections
        SCAN_RESULTS=$(tail -n 10 $LOG_FILE | grep 'Infected files')
        INFECTIONS=${SCAN_RESULTS##* }

        EMAILMESSAGE=`mktemp /tmp/virus-alert.XXXXX`
        echo "To: ${EMAIL}" >> ${EMAILMESSAGE}
        echo "From: ${EMAIL_FROM}" >> ${EMAILMESSAGE}
        echo "Subject: ${SUBJECT}" >> ${EMAILMESSAGE}
        echo "Importance: High" >> ${EMAILMESSAGE}
        echo "X-Priority: 1" >> ${EMAILMESSAGE}
        echo "****A Scan was carried out by FOLusername****" >> ${EMAILMESSAGE}

        if [ $AGGRESSIVE = 1 ]
        then
            echo -e "\n`tail -n $((10 + ($INFECTIONS*2))) $LOG_FILE`" >>
${EMAILMESSAGE}
        else
            echo -e "\n`tail -n $((10 + $INFECTIONS)) $LOG_FILE`" >>
${EMAILMESSAGE}
        fi
    fi
}
```

```

fi

    sendmail -t < ${EMAILMESSAGE}

fi
}

if [ $AGGRESSIVE = 1 ]
then
    /usr/bin/clamscan -ri --remove $SCAN_DIR >> $LOG_FILE
else
    /usr/bin/clamscan -ri $SCAN_DIR >> $LOG_FILE
fi

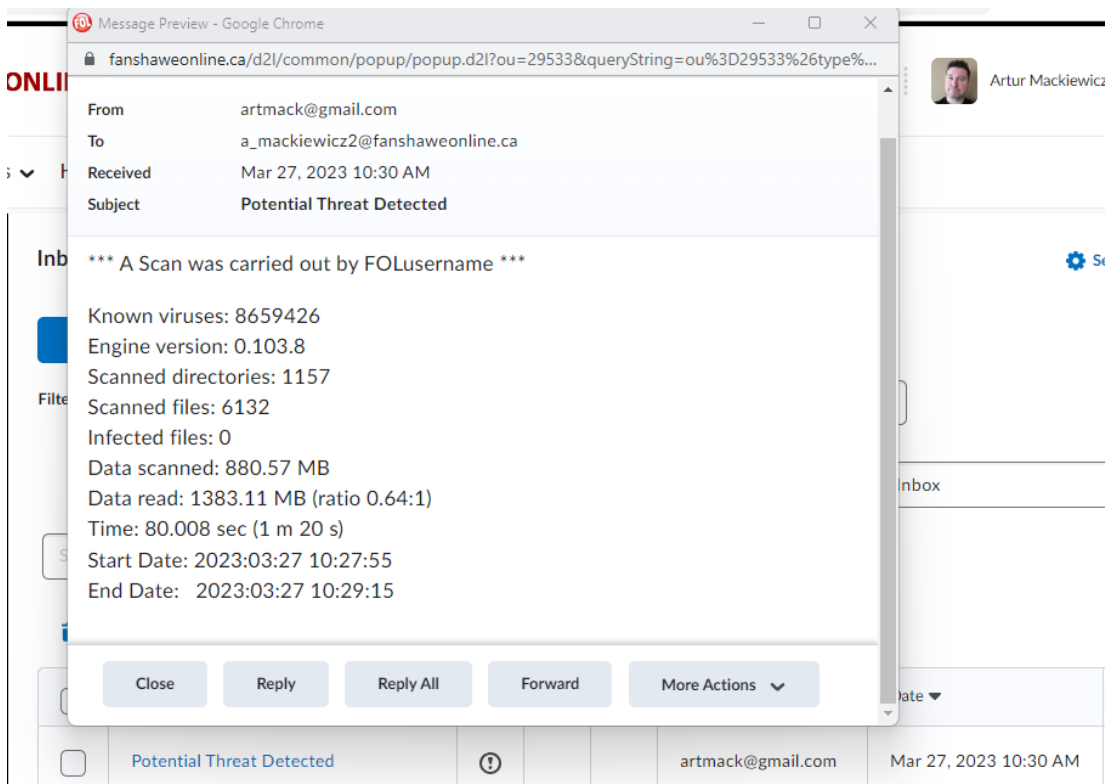
check_scan

```

Ensure that the script has execute permissions

Run the script manually. If you have no infections, there will not be an email that is generated. Adjust the script so that it sends out an email notification when **NO** threats are detected.

Run the script again and check to see if you have received the email notification



Slide 03:

- Take a screenshot showing the email that was generated by your script and place it into slide 03
- Show your FOL email account displaying the email you received from Gmail exactly as shown above

***** Shutdown the VM and take a snapshot called After Lab 10 *****