LAB5_MAC
AKOR JACOB TERUNGWA

Created	@March 6, 2025 6:34 PM
Attendance Required	☐

PART A: AppArmor

1: Explain How CIS Benchmarks Are Checked on an Endpoint Using a SIEM

First of all, it's important to define what CIS BENCHMARK is for a better understanding.

- **CIS Benchmarks**: CIS (Center for Internet Security) benchmarks are a set of best practices for securing IT systems and data. They provide guidelines for hardening operating systems, software, and network devices.

1. **How a SIEM Checks CIS Benchmarks**:
    - **Log Collection**: The SIEM collects logs from endpoints (e.g., Linux systems) using agents or syslog.
    - **Rule-Based Analysis**: The SIEM uses pre-defined rules to analyze logs and compare system configurations against CIS benchmarks.
    - **Compliance Reports**: The SIEM generates compliance reports highlighting deviations from the benchmarks.
    - **Example**: If a CIS benchmark recommends disabling root login via SSH, the SIEM checks the SSH configuration file (`/etc/ssh/sshd_config`) for `PermitRootLogin no` and alerts if the setting is incorrect.

2: Fulfill the MAC Section of the Latest CIS Benchmark for a Linux Distribution

- I visited the CIS Benchmarks website and downloaded the latest benchmark for Linux distribution (Ubuntu) at https://downloads.cisecurity.org/#/
- **I implemented MAC (Mandatory Access Control)**:
- Installed apparmor and apparmor-utils and ensured its enabled
    - sudo apt install apparmor apparmor-utils
    - sudo systemctl enable apparmor
    - sudo systemctl start apparmor

```
jacob@jacob-virtual-machine:~$ sudo apt install apparmor apparmor-utils
sudo systemctl enable apparmor
sudo systemctl start apparmor
[sudo] password for jacob:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apparmor is already the newest version (3.0.4-2ubuntu2.4).
apparmor set to manually installed.
The following additional packages will be installed:
  python3-apparmor python3-libapparmor
Suggested packages:
  vim-addon-manager
The following NEW packages will be installed:
  apparmor-utils python3-apparmor python3-libapparmor
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 170 kB of archives.
After this operation, 1.186 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-libapparmor amd64 3.0.4-2ubuntu2.4 [29,5 kB]
Get:2 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-apparmor all 3.0.4-2ubuntu2.4 [81,1 kB]
Get:3 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apparmor-utils all 3.0.4-2ubuntu2.4 [59,5 kB]
Fetched 170 kB in 2s (113 kB/s)
Selecting previously unselected package python3-libapparmor.
(Reading database ... 328164 files and directories currently installed.)
Preparing to unpack .../python3-libapparmor_3.0.4-2ubuntu2.4_amd64.deb ...
Unpacking python3-libapparmor (3.0.4-2ubuntu2.4) ...
Selecting previously unselected package python3-apparmor.
Preparing to unpack .../python3-apparmor_3.0.4-2ubuntu2.4_all.deb ...
Unpacking python3-apparmor (3.0.4-2ubuntu2.4) ...
Selecting previously unselected package apparmor-utils.
Preparing to unpack .../apparmor-utils_3.0.4-2ubuntu2.4_all.deb ...
Unpacking apparmor-utils (3.0.4-2ubuntu2.4) ...
```

```
Unpacking apparmor-utils (3.0.4-2ubuntu2.4) ...
Setting up python3-libapparmor (3.0.4-2ubuntu2.4) ...
Setting up python3-apparmor (3.0.4-2ubuntu2.4) ...
Setting up apparmor-utils (3.0.4-2ubuntu2.4) ...
Processing triggers for man-db (2.10.2-1) ...
Synchronizing state of apparmor.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apparmor
```

Then, I edited `/etc/default/grub` and modified `GRUB_CMDLINE_LINUX_DEFAULT` as the correct bootloader configuration

- sudo nano /etc/default/grub

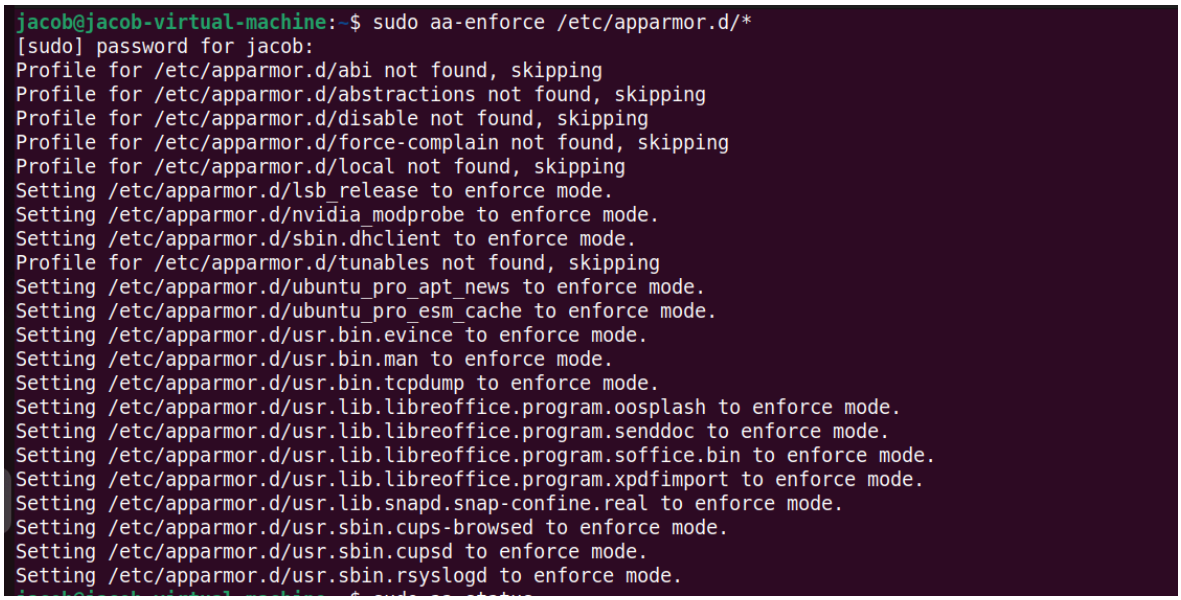
```
  GNU nano 6.2                                          /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US"
```

I updated the Grub and rebooted

- sudo update-grub
- sudo reboot

I needed **root** privileges to enforce AppArmor profiles for critical services and ensure all profiles are in "enforce" mode

- sudo aa-enforce /etc/apparmor.d/*



I checked AppArmor status:

- sudo aa-status

```
jacob@jacob-virtual-machine:~$ sudo aa-status
apparmor module is loaded.
62 profiles are loaded.
60 profiles are in enforce mode.
   /snap/snapd/20671/usr/lib/snapd/snap-confine
   /snap/snapd/20671/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
   /snap/snapd/23545/usr/lib/snapd/snap-confine
   /snap/snapd/23545/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
   /usr/bin/evince
   /usr/bin/evince-previewer
   /usr/bin/evince-previewer//sanitized_helper
   /usr/bin/evince-thumbnailer
   /usr/bin/evince//sanitized_helper
   /usr/bin/evince//snap_browsers
   /usr/bin/man
   /usr/lib/NetworkManager/nm-dhcp-client.action
   /usr/lib/NetworkManager/nm-dhcp-helper
   /usr/lib/connman/scripts/dhclient-script
   /usr/lib/cups/backend/cups-pdf
   /usr/lib/snapd/snap-confine
   /usr/lib/snapd/snap-confine//mount-namespace-capture-helper
   /usr/sbin/cups-browsed
   /usr/sbin/cupsd
   /usr/sbin/cupsd//third_party
   /{,usr/}sbin/dhclient
   docker-default
   libreoffice-oosplash
   libreoffice-senddoc
   libreoffice-soffice
   libreoffice-soffice//gpg
   libreoffice-xpdfimport
   lsb_release
   man_filter
```

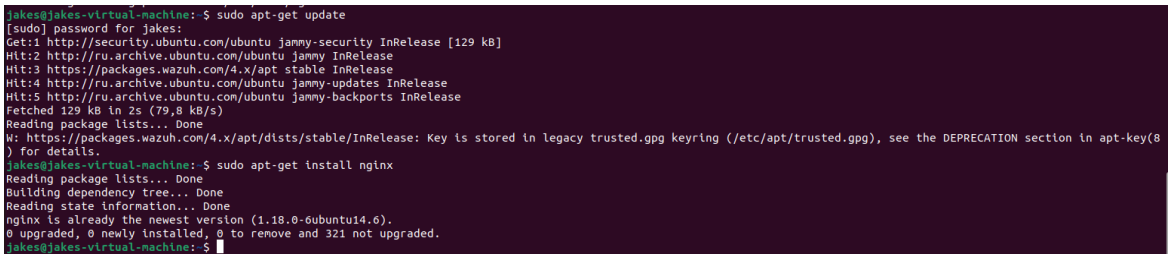
```
2 profiles are in complain mode.
   snap.code.code
   snap.code.url-handler
0 profiles are in kill mode.
0 profiles are in unconfined mode.
17 processes have profiles defined.
17 processes are in enforce mode.
   /usr/sbin/cups-browsed (1138)
   /usr/sbin/cupsd (987)
   /snap/firefox/5836/usr/lib/firefox/firefox (4861) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5077) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5113) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5121) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5167) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5596) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5629) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5721) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5951) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (5975) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (6127) snap.firefox.firefox
   /snap/firefox/5836/usr/lib/firefox/firefox (6200) snap.firefox.firefox
   /snap/snap-store/1216/usr/bin/snap-store (3454) snap.snap-store.ubuntu-software
   /snap/snapd-desktop-integration/253/usr/bin/snapd-desktop-integration (3036) snap.snapd-desktop-integration.snapd-desktop-integration
   /snap/snapd-desktop-integration/253/usr/bin/snapd-desktop-integration (3171) snap.snapd-desktop-integration.snapd-desktop-integration
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
0 processes are in mixed mode.
0 processes are in kill mode.
```

3: Configure a Webapp to Serve Static Files and Confine It with AppArmor

Setting Up a Webapp:

- I installed a web server (Nginx):
  - sudo apt-get update

- sudo apt-get install nginx



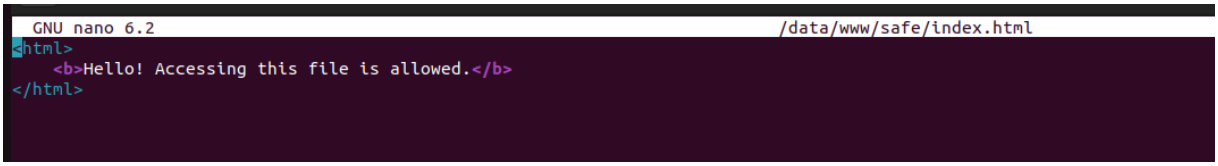
My Nginx server is operational, as the default server will run on port 80. I tested it in a browser by using my IP address as the URL: http://192.168.47.129:80. Then, I saw the default Nginx welcome page.

Create the directories from which the static files will be served.

- `sudo mkdir -p /data/www/safe`
- `sudo mkdir -p /data/www/unsafe`

Then, I added a file to the safe directory using nano:

```
sudo nano /data/www/safe/index.html
```



Similarly, I created another file in /data/www/unsafe named index.html, with the following contents:

sudo nano /data/www/unsafe/index.html

```
 GNU nano 6.2                                        /data/www/unsafe/index.html
<html>
    <b>Hello! Accessing this file is NOT allowed.</b>
</html>
```

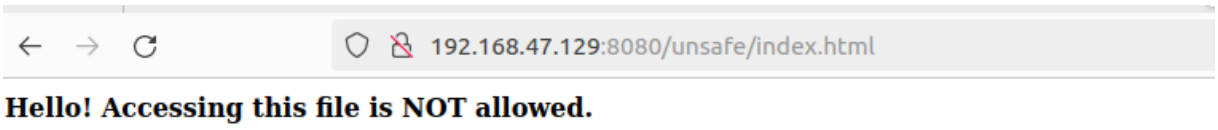
The nginx's configuration file is located at /etc/nginx/nginx.conf. I edited the file to create a new server that listens on port 8080 and serves files from /data/www

```
jakes@jakes-virtual-machine: ~
 GNU nano 6.2                                        /etc/nginx/nginx.conf *
user www-data;
worker_processes 4;
pid /run/nginx.pid;

events {
    worker_connections 768;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    gzip on;
    gzip_disable "msie6";

    include /etc/nginx/conf.d/*.conf;

    server {
        listen 8080;
        location / {
                root /data/www;
        }
    }
}
```

I saved the changes and loaded the new configuration by executing the following command:

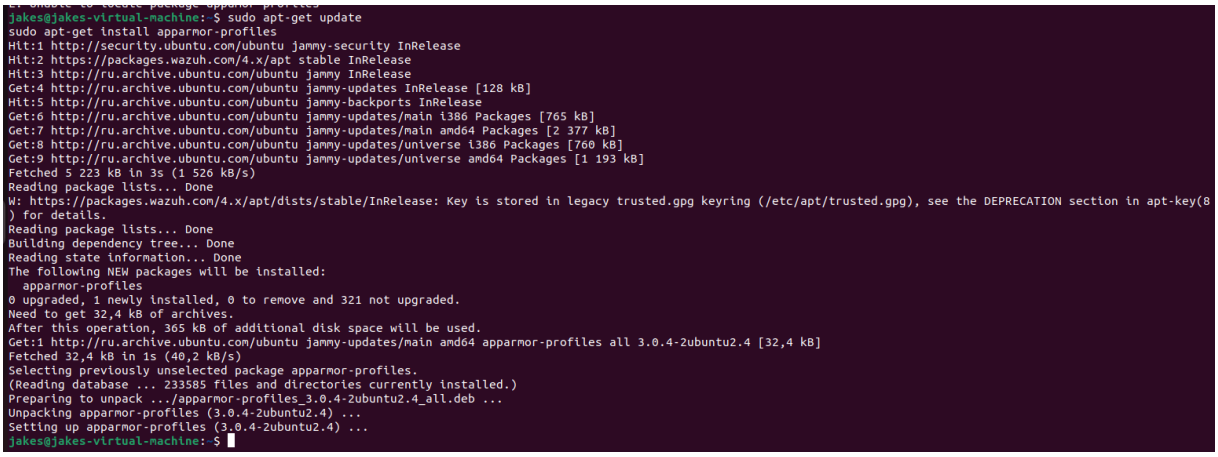
```
sudo nginx -s reload
```

At this point, since AppArmor has not yet been turned on for Nginx, I was able to visit both http://192.168.47.129:8080/safe/index.html and http://192.168.47.129:8080/unsafe/index.html



Created an AppArmor Profile for the Webapp:

- Generated a profile for Nginx:



Then, I listed all available profiles by executing this command:

sudo apparmor_status

```
jakes@jakes-virtual-machine:~$ sudo apparmor_status
apparmor module is loaded.
54 profiles are loaded.
46 profiles are in enforce mode.
   /snap/snapd/20671/usr/lib/snapd/snap-confine
   /snap/snapd/20671/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
   /usr/bin/evince
   /usr/bin/evince-previewer
   /usr/bin/evince-previewer//sanitized_helper
   /usr/bin/evince-thumbnailer
   /usr/bin/evince//sanitized_helper
   /usr/bin/evince//snap_browsers
   /usr/bin/man
   /usr/lib/NetworkManager/nm-dhcp-client.action
   /usr/lib/NetworkManager/nm-dhcp-helper
   /usr/lib/connman/scripts/dhclient-script
   /usr/lib/cups/backend/cups-pdf
   /usr/lib/snapd/snap-confine
   /usr/lib/snapd/snap-confine//mount-namespace-capture-helper
   /usr/sbin/cups-browsed
   /usr/sbin/cupsd
   /usr/sbin/cupsd//third_party
   /{,usr/}sbin/dhclient
   libreoffice-oosplash
   libreoffice-senddoc
   libreoffice-soffice
   libreoffice-soffice//gpg
   libreoffice-xpdfimport
   lsb_release
   man_filter
   man_groff
   nvidia_modprobe
   nvidia_modprobe//kmod
   rsyslogd
   snap-update-ns.firefox
   snap-update-ns.snap-store
   snap-update-ns.snapd-desktop-integration
   snap.firefox.firefox
   snap.firefox.geckodriver
   snap.firefox.hook.configure
   snap.firefox.hook.connect-plug-host-hunspell
   snap.firefox.hook.disconnect-plug-host-hunspell
```

```
   ping
   samba-bgqd
   smbd
   smbldap-useradd
   smbldap-useradd///etc/init.d/nscd
   syslog-ng
   syslogd
   traceroute
0 profiles are in kill mode.
0 profiles are in unconfined mode.
21 processes have profiles defined.
16 processes are in enforce mode.
   /usr/sbin/cups-browsed (978)
   /usr/sbin/cupsd (921)
   /usr/sbin/rsyslogd (836) rsyslogd
   /snap/firefox/3836/usr/lib/firefox/firefox (4036) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (4185) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (4206) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (4366) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (4409) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (4578) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (4580) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (5445) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (5569) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (5571) snap.firefox.firefox
   /snap/firefox/3836/usr/lib/firefox/firefox (5603) snap.firefox.firefox
   /snap/snapd-desktop-integration/253/usr/bin/snapd-desktop-integration (2215) snap.snapd-desktop-integration.snapd-desktop-integration
   /snap/snapd-desktop-integration/253/usr/bin/snapd-desktop-integration (2288) snap.snapd-desktop-integration.snapd-desktop-integration
5 processes are in complain mode.
   /usr/sbin/nginx (5380)
   /usr/sbin/nginx (5396)
   /usr/sbin/nginx (5397)
   /usr/sbin/nginx (5398)
   /usr/sbin/nginx (5399)
0 processes are unconfined but have a profile defined.
0 processes are in mixed mode.
0 processes are in kill mode.
jakes@jakes-virtual-machine:~$
```

Created a New AppArmor Profile for Nginx

`sudo apt-get install apparmor-utils`

```
jakes@jakes-virtual-machine:~$ sudo apt-get install apparmor-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apparmor-utils is already the newest version (3.0.4-2ubuntu2.4).
0 upgraded, 0 newly installed, 0 to remove and 321 not upgraded.
jakes@jakes-virtual-machine:~$
```

At this point, I was ready to start profiling the activities of Nginx. I used the aa-autodep command to create a new blank profile. The profile will be created in /etc/apparmor.d

```
cd /etc/apparmor.d/
sudo aa-autodep nginx

Once the profile was created, I used aa-complain to put the profile in complain mode.
sudo aa-complain nginx
```

```
jakes@jakes-virtual-machine:~$ cd /etc/apparmor.d/
sudo aa-autodep nginx
Profile for /usr/sbin/nginx already exists - skipping.
jakes@jakes-virtual-machine:/etc/apparmor.d$ sudo aa-complain nginx
Setting /usr/sbin/nginx to complain mode.
jakes@jakes-virtual-machine:/etc/apparmor.d$
```

I made some changes to the auto-generated file for it to work properly. Open the `/etc/apparmor.d/usr.sbin.nginx`

The new **capability** lines allow Nginx to start new processes. The **deny** rule allows us to block Nginx from accessing the /data/www/unsafe/ directory.

```
 GNU nano 6.2                                    /etc/apparmor.d/usr.sbin.nginx *
#include <tunables/global>

/usr/sbin/nginx {
  #include <abstractions/apache2-common>
  #include <abstractions/base>
  #include <abstractions/nis>

  capability dac_override,
  capability dac_read_search,
  capability net_bind_service,
  capability setgid,
  capability setuid,

  /data/www/safe/* r,
  deny /data/www/unsafe/* r,
  /etc/group r,
  /etc/nginx/conf.d/ r,
  /etc/nginx/mime.types r,
  /etc/nginx/nginx.conf r,
  /etc/nsswitch.conf r,
  /etc/passwd r,
  /etc/ssl/openssl.cnf r,
  /run/nginx.pid rw,
  /usr/sbin/nginx mr,
  /var/log/nginx/access.log w,
  /var/log/nginx/error.log w,
}
```

The AppArmor Nginx profile was ready, and I used the aa-enforce to put the profile in enforce mode.
Afterward, I reloaded all profiles and restarted Nginx to be sure that the latest changes were in effect.

I went back to the browser and visited http://192.168.47.129:8080/safe/index.html. I was able to see the page. Then visit http://192.168.47.129:8080/unsafe/index.html. I was able to see an error page as shown below. This proves that my profile is working as expected.

4: Explain How AppArmor Uses Default Profiles

1. **Default Profiles**:
   - AppArmor ships with default profiles for common applications (e.g., Nginx, Apache).
   - These profiles define what files and resources the application can access.
   - Example: The Nginx profile allows read access to `/var/www/html` but denies access to other directories.
2. **How It Works**:

- AppArmor enforces these profiles at the kernel level.
- If an application tries to access a resource not allowed by its profile, the action is blocked, and a log entry is created.

4. In a situation where your Webapp fails to start or misbehaves after the Apparmor profile has been enforced i.e AppArmor confinement, how would you rectify this? What steps would you take to troubleshoot this?

If the Nginx server fails to start after enforcing the profile, the profile does not include a permission that Nginx needs. One should check:

- The error text
- var/log/syslog
- /var/log/nginx/error.log

Then you have to modify your profile based on those errors

Step 1: Switch to Complain Mode

1. **Switch the Profile to Complain Mode**:
   - Temporarily switching the AppArmor profile from "enforce" mode to "complain" mode. This allows the application to run while logging violations:
     - sudo aa-complain /usr/sbin/nginx
2. **Restart the Web Application**:
   - Restart the web application (Nginx):
     - sudo systemctl restart nginx
3. **Verify the Application is Running**:
   - Check the status of the web application:
     - sudo systemctl status nginx

Step 2: Analyze AppArmor Logs

1. **Check AppArmor Logs**:
   - View the AppArmor logs to identify which resources or actions are being denied:
     - sudo cat /var/log/syslog | grep apparmor
2. **Identifying Blocked Resources**:

- Noting the resources (e.g., files, directories) and actions (e.g., read, write) that are being denied

Step 3: Update the AppArmor Profile

1. **Edit the Profile**:
   - Open the AppArmor profile for editing:
     - sudo nano /etc/apparmor.d/usr.sbin.nginx
2. **Add Necessary Permissions**:
   - Based on the logs, add the necessary permissions to the profile. For example:
     - Allow read access to a file:
       - /path/to/resource r,
       - /path/to/directory/** w,
3. **Save the Profile**

Step 4: Reload AppArmor

1. **Reload AppArmor**:
   - Reload AppArmor to apply the updated profile:
     - sudo systemctl reload apparmor

2. **Verify the Profile**:
   - Check the status of the profile:
     - sudo aa-status

Step 5: Switch Back to Enforce Mode

1. **Switch to Enforce Mode**:
   - Switch the profile back to "enforce" mode:
     - sudo aa-enforce /usr/sbin/nginx
2. **Restart the Web Application**:
   - Restart the web application:
     - sudo systemctl restart nginx

Step 6: Test the Web Application

1. **Test Access**:

- Test the web application to ensure it works as expected:

  curl http://localhost/
  curl
  http://localhost/safe/index.html
  curl
  http://localhost/unsafe/index.html

Save the updated profile and reload AppArmor:

- sudo systemctl reload apparmor

Step 8: Verify the Final Configuration

1. **Test Again**:
   - Repeat the tests to ensure the web application works as expected:
     - Access allowed directories (should work).
     - Access restricted directories (should be blocked).

Task 2: Deploy a Webapp, Stress Test, and Enable SELinux

Short Explanation of SELinux

- **SELinux**: (Security-Enhanced Linux) is a MAC system that enforces security policies at the kernel level.
- It provides fine-grained control over processes, files, and network resources.

How It Works:

- SELinux assigns labels (contexts) to files and processes.
- Policies define which contexts can interact with each other.
- If a process tries to access a resource not allowed by the policy, SELinux blocks the action.

2: Deploy a Webapp, Stress Test, and Enable SELinux

Deployed a Webapp:

I had successfully created and deployed an AppArmor Profile for a Webapp already in above task in Part A

To perform the stress test, I demonstrated :

- Installed a web server (Apache)

- sudo apt install apache2

```
jacob@jacob-virtual-machine:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 1.721 kB of archives.
After this operation, 7.137 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11,3 kB]
Get:2 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9.170 B]
Get:3 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.13 [1.348 kB]
Get:4 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.13 [165 kB]
Get:5 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.13 [89,0 kB]
Get:6 http://de.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.13 [97,9 kB]
Fetched 1.721 kB in 2s (1.011 kB/s)
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
(Reading database ... 328860 files and directories currently installed.)
Preparing to unpack .../0-libaprutil1-dbd-sqlite3_1.6.1-5ubuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../1-libaprutil1-ldap_1.6.1-5ubuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../2-apache2-bin_2.4.52-1ubuntu4.13_amd64.deb ...
```

```
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.Progress: [ 92%] [#####################################
#######.....]

Processing triggers for man-db (2.10.2-1) ...##################################################################################.........
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
```

Stress Test the Webapp:

- Used a tool like `ab` (Apache Benchmark) to stress test the web server:
  - ab -n 1000 -c 100 http://192.168.47.138/

```
jacob@jacob-virtual-machine:~$ ab -n 1000 -c 100 http://192.168.47.138/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.47.138 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Apache/2.4.52
Server Hostname:        192.168.47.138
Server Port:            80

Document Path:          /
Document Length:        10671 bytes

Concurrency Level:      100
Time taken for tests:   0.213 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      10945000 bytes
```

```
HTML transferred:       10671000 bytes
Requests per second:    4700.07 [#/sec] (mean)
Time per request:       21.276 [ms] (mean)
Time per request:       0.213 [ms] (mean, across all concurrent requests)
Transfer rate:          50236.54 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    2   3.4      1      14
Processing:     8   18   6.5     16      53
Waiting:        1   16   5.8     15      51
Total:          9   20   7.0     18      60

Percentage of the requests served within a certain time (ms)
  50%     18
  66%     21
  75%     22
  80%     24
  90%     28
  95%     33
  98%     42
  99%     48
 100%     60 (longest request)
```

From the attached output, the performance metric is as follows:

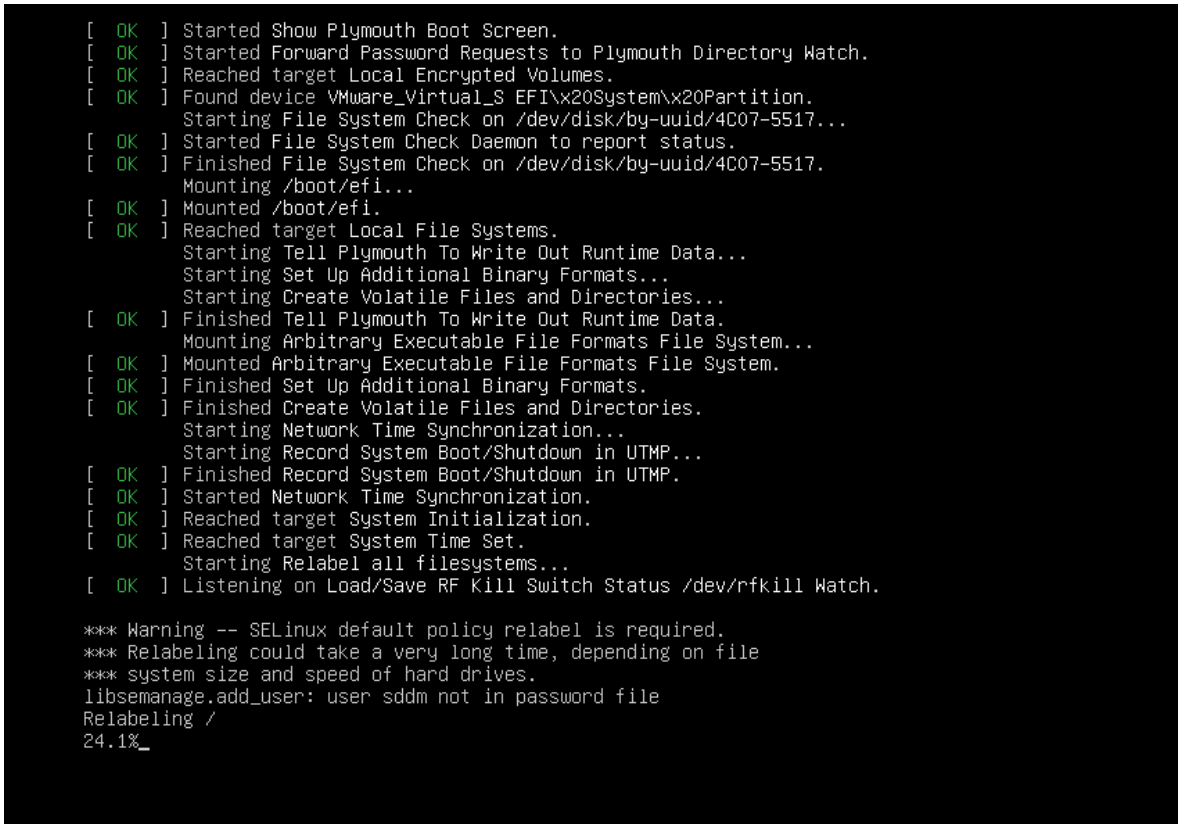
- Time taken for tests: 0.213 seconds
- Complete requests: 1000
- Failed requests: 0
- Total transferred: 10945000 bytes
- Requests per second: 4700.07 [#/sec] (mean)
- Time per request: 21.276 [ms] (mean)
- Time per request: 0.213 [ms] (mean, across all concurrent requests)
- Transfer rate: 50236.54 [Kbytes/sec] received

Installed and Enabled SELinux:

- sudo apt install selinux-basics selinux-policy-default auditd
- sudo selinux-activate

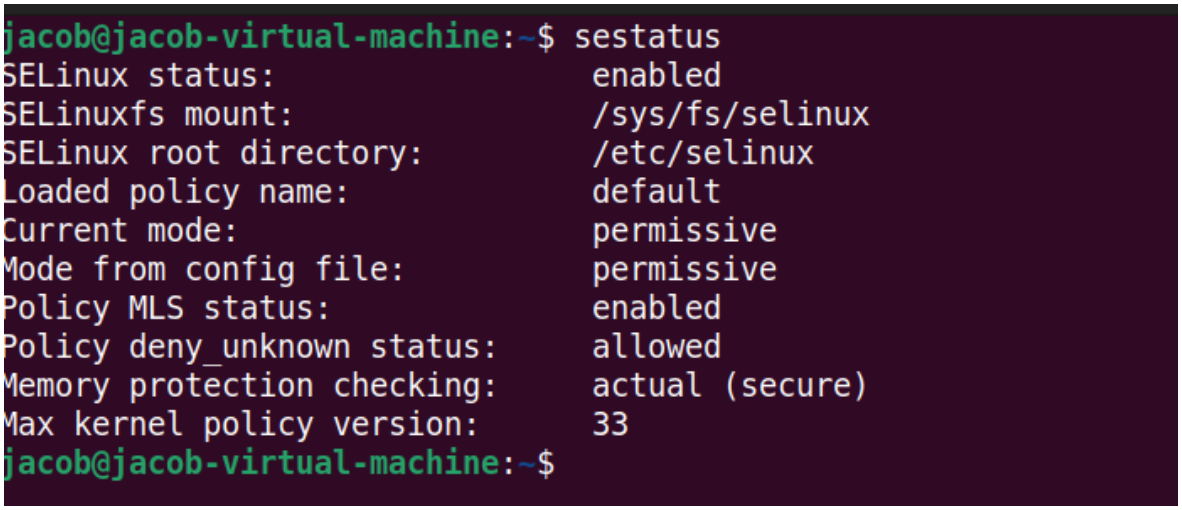
```
jacob@jacob-virtual-machine:~$ sudo selinux-activate
Activating SE Linux
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.8.0-52-generic
Found initrd image: /boot/initrd.img-6.8.0-52-generic
Found linux image: /boot/vmlinuz-6.5.0-18-generic
Found initrd image: /boot/initrd.img-6.5.0-18-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
SE Linux is activated.  You may need to reboot now.
jacob@jacob-virtual-machine:~$
```

- sudo reboot



Verified SELinux status:

- sestatus



I created a custom policy for the web server:

1. **Identify Denials:**

- I looked for `AVC (Access Vector Cache)` denials in the logs.

```
time->Tue Mar 11 15:56:44 2025
type=PROCTITLE msg=audit(1741697804.107:384): proctitle=2F7573722F6C6962657865632F706F6C6B697464002D2D6E6F2D6465627567
type=SYSCALL msg=audit(1741697804.107:384): arch=c000003e syscall=9 success=yes exit=123396858777600 a0=0 a1=621 a2=1 a3=2 items=0 ppid=1 pid=697 auid=4294967295 uid=0 gid=0 euid=0
suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="polkitd" exe="/usr/libexec/polkitd" subj=system_u:system_r:policykit_t:s0 key=(null)
type=AVC msg=audit(1741697804.107:384): avc:  denied  { map } for  pid=697 comm="polkitd" path="/usr/share/locale-langpack/en/LC_MESSAGES/accounts-service.mo" dev="sda3" ino=455655
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:object_r:usr_t:s0 tclass=file permissive=1
----
time->Tue Mar 11 15:57:38 2025
type=PROCTITLE msg=audit(1741697858.491:386): proctitle="/usr/bin/gnome-shell"
type=SYSCALL msg=audit(1741697858.491:386): arch=c000003e syscall=9 success=yes exit=61229040992256 a0=37afff3d0000 a1=10000 a2=5 a3=32 items=0 ppid=2156 pid=2339 auid=1000 uid=1000
 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=(none) ses=3 comm="gnome-shell" exe="/usr/bin/gnome-shell" subj=unconfined_u:unconfined_r:unconfined_t:s0
-s0:c0.c1023 key=(null)
type=AVC msg=audit(1741697858.491:386): avc:  denied  { execmem } for  pid=2339 comm="gnome-shell" scontext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 tcontext=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 tclass=process permissive=1
jakes@jakes-virtual-machine:~$
```

2. **I generated a Policy Module:**

- Used `audit2allow` to create a policy module from the denials:
  - sudo grep httpd /var/log/audit/audit.log | audit2allow -M mywebapp

This command creates two files:

- `mywebapp.te`: Type Enforcement file (policy source).

```
 GNU nano 6.2                                                    mywebapp.te

module mywebapp 1.0;

require {
        type httpd_t;
        type httpd_sys_rw_content_t;
        class dir getattr;
}

#============= httpd_t ==============

#!!!! This avc can be allowed using the boolean 'httpd_builtin_scripting'
allow httpd_t httpd_sys_rw_content_t:dir getattr;
```

- `mywebapp.pp`: Compiled policy module.

```
jakes@jakes-virtual-machine: ~
 GNU nano 6.2                                                    mywebapp.pp
^@^@^@^@^A^@^@^@^P^@^@^@◆◆◆|◆^O^@^@^@SE Linux Module^B^@^@^@^T^@^@^@^A^@^@^@^H^@^@^@^@^@^@^@^H^@^@^@mywebapp^C^@^@^@1.0@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^A^@^@^@^A^@^@^@^C^@
```

3. **I reviewed and edited the `.te` File:**

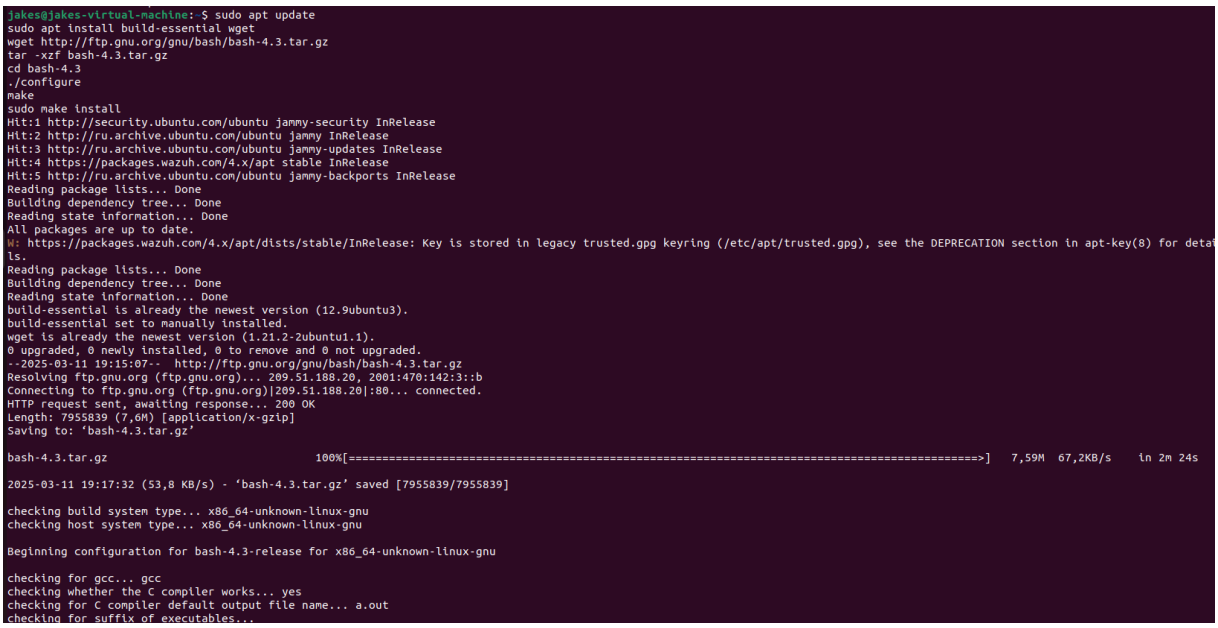
```
 GNU nano 6.2                                                    mywebapp.te *
module mywebapp 1.0;
require {
    type httpd_t;
    type default_t;
    class file { read };
}
allow httpd_t default_t:file { read };
```

Bonus Task

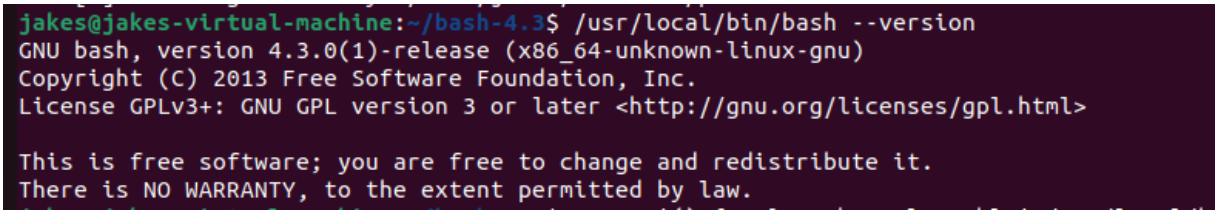
Step 1: I set Up a Vulnerable Environment on Ubuntu

Installed a Vulnerable Version of Bash:

sudo apt update
sudo apt install build-essential wget
wget
http://ftp.gnu.org/gnu/bash/bash-4.3.tar.gz
tar -xzf bash-4.3.tar.gz
cd bash-4.3
./configure
make
sudo make install



Checked the installed Bash version to be sure the installation is complete:



Replicated the Shellshock Attack

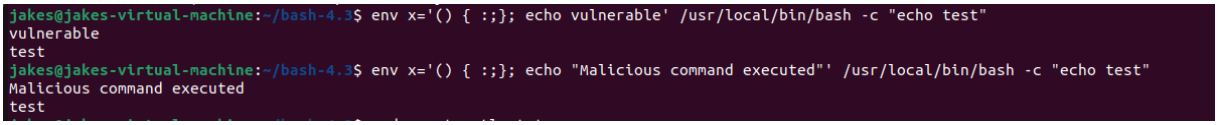
Tested for Vulnerability: I ran the following command to check if the system is vulnerable:

- env x='() { :;}; echo vulnerable' /usr/local/bin/bash -c "echo test”

To exploit the Vulnerability:

I used the vulnerability to execute a malicious command:

- env x='() { :;}; echo "Malicious command executed"' /usr/local/bin/bash -c "echo test”



To mitigate the Shellshock Attack

I started and enabled the AppAmor

sudo systemctl start apparmor
sudo systemctl enable apparmor

Then, I generated a profile for Bash using the command:

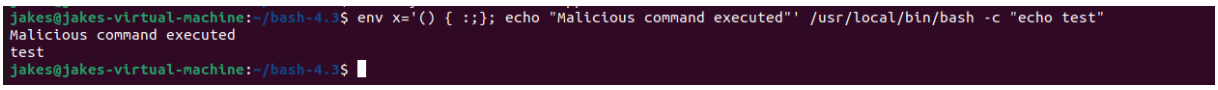
sudo aa-genprof /usr/local/bin/bash

I proceeded by setting the profile to enforce mode:

sudo aa-enforce /usr/local/bin/bash

Lastly, I tested the Exploit Again by attempting the Shellshock exploit

env x='() { :;}; echo "Malicious command executed"' /usr/local/bin/bash -c "echo test”



Mitigating Using SELinux

I Installed SELinux and related tools:

```
jakes@jakes-virtual-machine:~$ sudo apt update
sudo apt install selinux-basics selinux-policy-default auditd
[sudo] password for jakes:
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 https://packages.wazuh.com/4.x/apt stable InRelease
Hit:4 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
W: https://packages.wazuh.com/4.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for detai
ls.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
auditd is already the newest version (1:3.0.7-1build1).
selinux-basics is already the newest version (0.5.8).
selinux-policy-default is already the newest version (2:2.20210203-10).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jakes@jakes-virtual-machine:~$
```

I activated SELinux

`sudo selinux-activate`

```
jakes@jakes-virtual-machine:~$ sudo selinux-activate
Activating SE Linux
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.8.0-52-generic
Found initrd image: /boot/initrd.img-6.8.0-52-generic
Found linux image: /boot/vmlinuz-6.5.0-18-generic
Found initrd image: /boot/initrd.img-6.5.0-18-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
SE Linux is activated.  You may need to reboot now.
```

After this, I rebooted using sudo reboot

After rebooting, I had to set SELinux to enforcing mode:

sudo setenforce 1

References:

https://www.digitalocean.com/community/tutorials/how-to-create-an-apparmor-profile-for-nginx-on-ubuntu-14-04#step-four-create-a-new-apparmor-profile-for-nginx

https://documentation.wazuh.com/4.9/deployment-options/virtual-machine/virtual-machine.html