NSPJ24 Access Control

Overview

AppArmor is a a name-based (path-based) Mandatory Access Control (MAC) based on Linux Security Module (LSM). It can restrict the permissions of an application through path entries and capability entries.

In this project, you are required to create an AppArmor profile controls how Nginx behaves, as well as restrict a container's access to resources. Note that this homework is run on Ubuntu 22.04

Here’ s some common apparmor APIs

1. `aa-easyprof`: provides an easy to use interface for AppArmor policy generation.
2. `aa-logprof`: an interactive tool used to review AppArmor generated messages and update AppArmor security profiles.

Task 1. Permission access

In this section, we will be creating AppArmor profiles to allow `safe/index.html` and not allowing `unsafe/index.html`

We will be using Nginx to apply this task.

Task 1.1 Setting up Nginx

1. Install Nginx
2. Move `safe/index.html` and `unsafe/index.html` to the root folder(or any folder as long as you remember to include it within your config file) of your Nginx
3. Remember to provide a screenshot to prove that this section works

Task 1.2 Creating AppArmor profile

1. Create an AppArmor profile which allows Nginx to: (Note that the profile name should be **nspj-nginx_studentID)**
   a. Read `safe/index.html`
   b. Restrict Ngnix to read `unsafe/index.html`
   c. Entries required for Nginx to work properly
2. If your profile works, you (any user) should be able to see the contents of `safe/index.html` while you should not be able to see the contents of `unsafe/index.html`
3. Remember to provide a screenshot to prove that this section works.

Task 2. Restrict a Container's Access to Resources

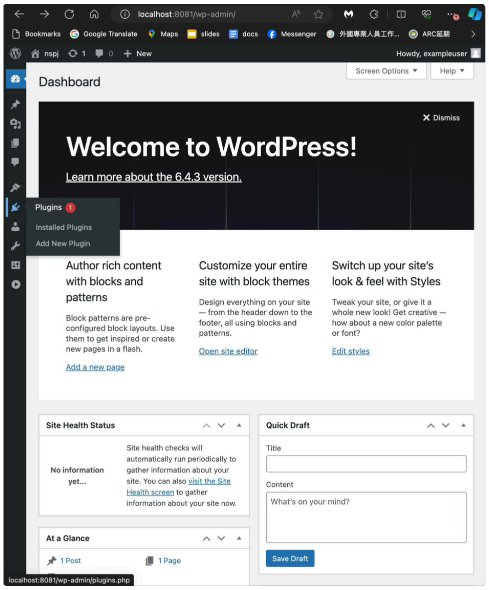
Background of this task is that since WordPress and its plugins run as PHP. This means an attacker could upload their own malicious plugin to start a shell on WordPress by simply sending a request to the PHP resource to run the malicious code and spin up the shell. The objective of this task is to apply AppArmor to protect WordPress within a docker. The provided .zip file contains the files required to setup task 2.

Task 2.1 Setting up the container

1. Run the following command
   a. `sudo apt-get update && apt-get install docker-compose`
   b. `docker build -t nspj:latest .`

    c. `docker-compose up`

2. After logging in using the username/password provided from within the `docker-compose.yml` file, try to install any plugin from the left-hand navigation pane to install a plugin. (You should be able to install anything from within the plugin page)



Dashboard

Welcome to WordPress!

Learn more about the 6.4.3 version.

Dismiss

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.

At a Glance

1 Post 1 Page

Quick Draft

Title

Content

What's on your mind?

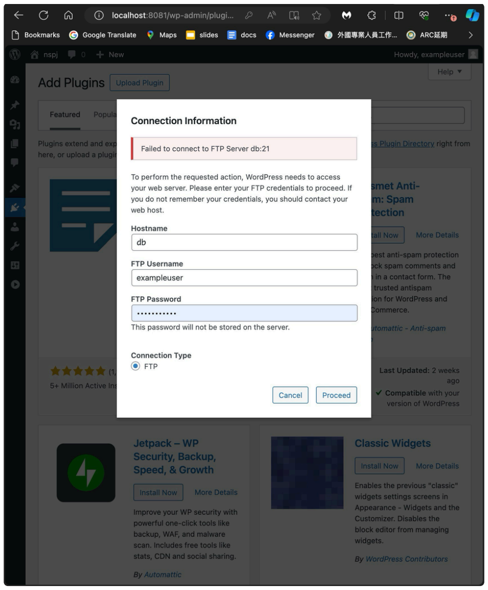
Save Draft

3. Remember to provide a screenshot to prove that this section works

Task 2.2 Creating AppArmor profile

Here, we will create a profile which prevents malicious themes and plugins from being uploaded and installed on our WordPress instance.

1. Create an AppArmor profile named **nspj-docker_studentID**
2. Attach the profile into `docker-compose.yml`
3. The profile should be able to do:
    a. Allow docker to be hosted normally and wordpress to work as intended.
    b. Restricting everything under ./wp-content **except** uploads directory(this is used for media usage)
    c. No network access (ping, traceroute, etc)



Connection Information

Failed to connect to FTP Server db:21

To perform the requested action, WordPress needs to access your web server. Please enter your FTP credentials to proceed. If you do not remember your credentials, you should contact your web host.

Hostname: db

FTP Username: exampleuser

FTP Password: ••••••••••••

This password will not be stored on the server.

Connection Type

● FTP

Cancel Proceed

Jetpack – WP Security, Backup, Speed, & Growth

Improve your WP security with powerful one-click tools like backup, WAF, and malware scan. Includes free tools like stats, CDN and social sharing.

By Automattic

Classic Widgets

Enables the previous "classic" widgets settings screens in Appearance - Widgets and the Customizer. Disables the block editor from managing widgets.

By WordPress Contributors

4. Remember to provide a screenshot to prove that this section works

Report

You should include these on your report:

- Task 1 & 2
  - Before and after setting Nginx profile
  - Step-by-step on you did to setup AppArmor profiles.
- What you did to test whether your profile do work properly
  - Aside from the required tasks, see if you can design an interesting way to test your works. (Might get extra points)
- Describe difficulties (if any) you encountered in the implementations and how you have addressed them.

Submission

Your submission should include the requirement of each task:

1. `nspj-nginx_studentID` and `nspj-docker_studentID`
2. `report_student` in PDF

Please zip the entire files directly, do not zip a folder.

Pay attention to the filename format, else we will minus 5 points

If you have any questions, do email TA stevanhandi.ee11@nycu.edu.tw

nspj_hw2.zip