Assignments 〉 9.5 Assignment: Securing Web Applications with Docker and Ansible

9.5 Assignment: Securing Web Applications with Docker and Ansible

▼ Hide Assignment Information

Instructions

Estimated Effort Required: 10 hours

Learning Outcome: Students will gain experience in deploying and securing containers using automation tools.

Assignment Goal: You create an Ansible playbook and dockerfiles that can be used to secure a vulnerable web application running inside of a container.

Preface: The following words should be interpreted as per RFC 2119 for this assignment: must, must not, required, shall, shall not, should, should not, recommended, may, and optional.

Environment: You will need two virtual machines; one running Ansible that will act as your administrative host and another Ubuntu virtual machine that will act as a host for Docker containers. **If your home machine is not capable of running two virtual machines, please let me know and I'll create VMs for you on RIT Infrastructure similar to what was used in the last assignment.**

Requirements

1. Install Ansible on one virtual machine. Make sure that you download it from the Ansible Github repository; do not apt-get install it.
2. Install openssh-server and Python on the second virtual machine. Enable openssh-server.
3. Create an Ansible playbook which installs Docker and deploys two containers that meet the following criteria.
    a. The first container:
        i. Is based off of Docker Hub's Ubuntu 18.04 image.
        ii. Runs Apache and PHP.
        iii. Has the following script named vuln.php in the Apache webroot directory:

```
<?php echo “Hello “ . $_GET[‘name’]; ?>
```

Note: This script is vulnerable to cross-site scripting through the name parameter.

iv. Has directory listings turned off.

v. Has server tokens and signatures turned off.

vi. Has HTTPS enabled with a self-signed certificate

1. Note: You may want to look into openssl's -subj functionality.

vii. Exposes the container's port 80 on the host port 8080.

viii. Exposes the container's port 443 on the host port 8443.

e. The second container

i. Runs ModSecurity with the OWASP core rule set.

1. You may make use of any ModSecurity containers you find on Docker Hub or configure it yourself.

ii. Acts as a proxy, forwarding requests that pass ModSecurity checks on to the first container listening on port 8080.

iii. Exposes modsecurity on the Ubuntu host's port 80.

vi. Both containers run in Docker when the VM boots.

Deliverables

1. A zip file containing

a. An ansible playbook that meets the requirements above

b. Any supporting files needed for the Ansible playbook to execute properly

c. A README for properly executing your playbook, if necessary.

Grading Method

1. This assignment will be graded by:

a. Executing your Ansible playbook on an Ubuntu 18.04 VM with only Python and openssl-server configured to run.

b. Making an HTTPS request to port 8443 for /vuln.php to test your HTTPS configuration with the name parameter set to *Rob*
c. Making an HTTP request to your vuln.php on port 8080 with the name parameter set to <script>alert('xss');</script> to verify that your script is vulnerable. This cross-site scripting attack should be successful.
d. Repeating step C against port 80 to verify your ModSecurity configuration. The attack should fail.

Due Date

Apr 20, 2020 11:55 PM

Submit Assignment

Files to submit *

(0) file(s) to submit

IMPORTANT: After uploading, you must click Submit to complete the submission.

If you are not told on the next screen: ** FILE SUBMISSION SUCCESSFUL **, then your assignment was not submitted and no file has been saved for your instructor.

Add a File

Comments

Paragraph

Font Family

Font Size

Submit

Cancel