**Ansible Playbooks**

1. Connect to your jump box, and connect to the Ansible container in the box.
   * If you stopped your container or exited it in the last activity, find it again using docker container list -a.
2. root@Red-Team-Web-VM-1:/home/RedAdmin# docker container list -a
3. CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4. Exited (0) 2 minutes ago hardcore\_brown

a0d78be636f7 cyberxsecurity/ansible:latest "bash" 3 days ago

* + Start the container again using docker start [container\_name].

root@Red-Team-Web-VM-1:/home/RedAdmin# docker start hardcore\_brown

hardcore\_brown

* + Get a shell in your container using docker attach [container\_name].

root@Red-Team-Web-VM-1:/home/RedAdmin# docker attach hardcore\_brown

root@1f08425a2967:~#

1. Create a YAML playbook file that you will use for your configuration.

root@1f08425a2967:~# nano /etc/ansible/pentest.yml

The top of your YAML file should read similar to:

---

- name: Config Web VM with Docker

hosts: web

become: true

tasks:

* Use the Ansible apt module to install docker.io and python3-pip: **Note:** update\_cache must be used here, or docker.io will not install. (this is the equivalent of running apt update)
* - name: docker.io
* apt:
* update\_cache: yes
* name: docker.io
* state: present
* - name: Install pip3
* apt:
* force\_apt\_get: yes
* name: python3-pip

state: present

Note: update\_cache: yes is needed to download and install docker.io

* Use the Ansible pip module to install docker:
* - name: Install Python Docker module
* pip:
* name: docker

state: present

Note: Here we are installing the Python Docker Module, so Ansible can then utilize that module to control docker containers. More about the Python Docker Module [HERE](https://docker-py.readthedocs.io/en/stable/)

* Use the Ansible docker-container module to install the cyberxsecurity/dvwa container.
  + Make sure you publish port 80 on the container to port 80 on the host.
* - name: download and launch a docker web container
* docker\_container:
* name: dvwa
* image: cyberxsecurity/dvwa
* state: started
* restart\_policy: always

published\_ports: 80:80

NOTE: restart\_policy: always will ensure that the container restarts if you restart your web vm. Without it, you will have to restart your container when you restart the machine.

You will also need to use the systemd module to restart the docker service when the machine reboots. That block looks like this:

- name: Enable docker service

systemd:

name: docker

enabled: yes

1. Run your Ansible playbook on the new virtual machine.

Your final playbook should read similar to:

---

- name: Config Web VM with Docker

hosts: webservers

become: true

tasks:

- name: docker.io

apt:

force\_apt\_get: yes

update\_cache: yes

name: docker.io

state: present

- name: Install pip3

apt:

force\_apt\_get: yes

name: python3-pip

state: present

- name: Install Docker python module

pip:

name: docker

state: present

- name: download and launch a docker web container

docker\_container:

name: dvwa

image: cyberxsecurity/dvwa

state: started

published\_ports: 80:80

restart\_policy: always

- name: Enable docker service

systemd:

name: docker

enabled: yes

* Running your playbook should produce an output similar to the following:
* root@1f08425a2967:~# ansible-playbook /etc/ansible/pentest.yml
* PLAY [Config Web VM with Docker] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* ok: [10.0.0.6]
* TASK [docker.io] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* [WARNING]: Updating cache and auto-installing missing dependency: python-apt
* changed: [10.0.0.6]
* TASK [Install pip3] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* changed: [10.0.0.6]
* TASK [Install Docker python module] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* changed: [10.0.0.6]
* TASK [download and launch a docker web container] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* changed: [10.0.0.6]
* PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.0.0.6 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

1. To test that DVWA is running on the new VM, SSH to the new VM from your Ansible container.
   * SSH to your container:
2. root@1f08425a2967:~# ssh sysadmin@10.0.0.6
3. Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1027-azure x86\_64)
4. \* Documentation: https://help.ubuntu.com
5. \* Management: https://landscape.canonical.com
6. \* Support: https://ubuntu.com/advantage
7. System information as of Mon Jan 6 20:01:03 UTC 2020
8. System load: 0.01 Processes: 122
9. Usage of /: 9.9% of 28.90GB Users logged in: 0
10. Memory usage: 58% IP address for eth0: 10.0.0.6
11. Swap usage: 0% IP address for docker0: 172.17.0.1
12. 18 packages can be updated.
13. 0 updates are security updates.

Last login: Mon Jan 6 19:33:51 2020 from 10.0.0.4

* + Run curl localhost/setup.php to test the connection. If everything is working, you should get back some HTML from the DVWA container.

ansible@Pentest-1:~$ curl localhost/setup.php

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Setup :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\*</title>

<link rel="stylesheet" type="text/css" href="dvwa/css/main.css" />

<link rel="icon" type="\image/ico" href="favicon.ico" />

<script type="text/javascript" src="dvwa/js/dvwaPage.js"></script>

</head>