Generate a public key: ssh-keygen

Run cat ~/.ssh/id_rsa.pub to display your id_rsa.pub key:

Use your command line to SSH to the VM for administration. Windows users should use GitBash

The command to connect is ssh sysadmin@JumpBox-Public-IP

Command sudo -l : admin user has full sudo permissions without requiring a password.

Creating a container

Sudo apt install docker.io Start by installing docker.io on your Jump box.

Run sudo systemctl status docker

Once Docker is installed, pull the container cyberxsecurity/ansible

Run docker run -ti cyberxsecurity/ansible:latest bash to start the container.

Run exit to quit

launch a new VM from the Azure portal that could only be accessed using a new SSH key from the container running inside your jump box

Run docker images to view your image.

docker run -it cyberxsecurity/ansible /bin/bash to start your container and connect to it

Run cat .ssh/id_rsa.pub to display your public key.

fter your VM launches, test your connection using ssh from your jump box Ansible container.

Locate the Ansible config file and hosts file.

nano /etc/ansible/hosts

Uncomment the [webservers] header line

```
[webservers]

      ## alpha.example.org
```

```
## beta.example.org

## 192.168.1.100

## 192.168.1.110

10.0.0.6 ansible_python_interpreter=/usr/bin/python3

                    10.0.0.7 ansible_python_interpreter=/usr/bin/python3
```

Open the file with nano /etc/ansible/ansible.cfg and scroll down to the remote_user option

Uncomment the remote_user line and replace root with your admin username using this format: remote_user = sysadmin

Test an Ansible connection using the appropriate Ansible command.

ansible -m ping all

results:

```
10.0.0.5 | SUCCESS => {

"changed": false,

"ping": "pong"

}

10.0.0.6 | SUCCESS => {

            "changed": false,

            "ping": "pong"

}
```

Ansible playbook that installed Docker and configure a VM with the DVWA web application.

connect to the Ansible container in the box

docker container list -a

docker start [container_name].

docker attach [container_name]

Running your playbook should produce an output similar to the following

ansible-playbook /etc/ansible/pentest.yml

sudo docker start your_container_name

sudo docker attach your_container_name

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

Nano /etc/ansible/DVWA.yml

Example: `check ansible playbook on ansiblevm-Web2.yml`
To test that DVWA is running on the new VM, SSH to the web2 from your Ansible container(JumBox)

ssh `sysadmin@10.0.0.6`

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

```
## beta.example.org

## 192.168.1.100

## 192.168.1.110

10.0.0.6 ansible_python_interpreter=/usr/bin/python3

10.0.0.7 ansible_python_interpreter=/usr/bin/python3

10.0.0.8 ansible_python_interpreter=/usr/bin/python3

# If you have multiple hosts following a pattern you can specify

# them like this:
```

Test your Ansible configuration with the Ansible `ping` command.

- Run `ansible -m ping all`

Run ansible-playbook your-playbook.yml

· Run `ssh ansible@10.0.0.7`

· Run `curl localhost/setup.php`

SSH

Secure Shell sets up an encrypted connection between two machines. Commands given on the first machine are executed on the second machine and output from the second machine is sent back to the first machine.

The result is the ability to control a remote machine using the command line while keeping all your actions private from any would be attacker or snooper.

ssh-keygen

ssh `-i` mykey.pub `admin@10.10.0.4`

Elkserver commands:

Sudo docker container start [container name]

sudo docker container attach [container name]

to run the playbook command: root@0e52a6c7f577:/etc/ansible# ansible-playbook roles/filebeat-playbook.yml

alpha.example.org

beta.example.org

192.168.1.100

192.168.1.110

10.0.0.5 ansible_python_interpreter=/usr/bin/python3

10.0.0.6 ansible_python_interpreter=/usr/bin/python3

[Elk]

10.1.0.4 ansible_python_interpreter=/usr/bin/python3