**Automated ELK Stack Deployment**

The files in this repository were used to configure the network depicted below.

Diagram

Description automatically generated

These files have been tested and used to generate a live ELK deployment on Azure. They can be used to either recreate the entire deployment pictured above. Alternatively, select portions of the playbook file may be used to install only certain pieces of it, such as Filebeat.

Playbook 1: pentest.yml

---

- 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

- name: Enable docker service

systemd:

name: docker

enabled: yes

**Playbook 2: install-elk.yml**

---

- name: Configure Elk VM with Docker

hosts: elkservers

remote\_user: elk

become: true

tasks:

# Use apt module

- name: Install docker.io

apt:

update\_cache: yes

force\_apt\_get: yes

name: docker.io

state: present

# Use apt module

- name: Install python3-pip

apt:

force\_apt\_get: yes

name: python3-pip

state: present

# Use pip module (It will default to pip3)

- name: Install Docker module

pip:

name: docker

state: present

# Use command module

- name: Increase virtual memory

command: sysctl -w vm.max\_map\_count=262144

# Use sysctl module

- name: Use more memory

sysctl:

name: vm.max\_map\_count

value: 262144

state: present

reload: yes

# Use docker\_container module

- name: download and launch a docker elk container

docker\_container:

name: elk

image: sebp/elk:761

state: started

restart\_policy: always

# Please list the ports that ELK runs on

published\_ports:

- 5601:5601

- 9200:9200

- 5044:5044

#### Playbook 3: filebeat-playbook.yml

---

- name: installing and launching filebeat

hosts: webservers

become: yes

tasks:

- name: download filebeat deb

command: curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.0-amd64.deb

- name: install filebeat deb

command: dpkg -i filebeat-7.4.0-amd64.deb

- name: drop in filebeat.yml

copy:

src: /etc/ansible/files/filebeat-config.yml

dest: /etc/filebeat/filebeat.yml

- name: enable and configure system module

command: filebeat modules enable system

- name: setup filebeat

command: filebeat setup

- name: Start filebeat service

command: service filebeat start

**This document contains the following details:**

1. Description of the Topology
2. Access Policies
3. ELK Configuration
   * Beats in Use
   * Machines Being Monitored
4. How to Use the Ansible Build
5. **Description of the Topology**

The main purpose of this network is to expose a load-balanced and monitored instance of DVWA, the D\*mn Vulnerable Web Application.

Load balancing ensures that the application will be highly available, in addition to restricting access to the network.

* Load balancers protect application availability, allowing client requests to be shared across a number of servers
* The advantage of a Jump Box is that it minimises the attack surface by ensuring remote connections to the cloud network come through a single VM. Additionally, remote connections to the Jump Box can be monitored easily to identify unusual remote connections.

Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the configuration and system files.

* Filebeat is used to monitor logs files
* Metricbeat is used to collect operating system and service statistics from monitored VMs

The configuration details of each machine may be found below.

| **Name** | **Function** | **IP Address** | **Operating System** |
| --- | --- | --- | --- |
| Jump Box | Gateway | 10.0.0.4 | Linux |
| Web-1 | DVWA | 10.0.0.5 | Linux |
| Web-2 | DVWA | 10.0.0.6 | Linux |
| Web-3 | DVWA | 10.0.0.8 | Linux |
| Elk-1 | ELK | 10.1.0.4 | Linux |

1. **Access Policies**

The machines on the internal network are not exposed to the public Internet.

Only the Jump Box machine can accept SSH connections from the Internet. Access to this machine is only allowed from the following IP addresses:

* 184.145.129.160

Machines within the network can only be accessed by the Jump Box.

* The Jump Box can access the ELK VM **ELK-VM** using SSH. The Jump Box's IP address is 10.0.0.4

A summary of the access policies in place can be found in the table below.

| **Name** | **Publicly Accessible** | **Allowed IP Addresses** | **Allowed Ports** |
| --- | --- | --- | --- |
| Jump Box | Yes (SSH) | 184.145.129.160 | 22 |
| Web-1 | Yes (HTTP) | 20.213.240.243 | 80 |
| Web-2 | Yes (HTTP) | 20.213.240.243 | 80 |
| Web-3 | Yes (HTTP) | 20.213.240.243 | 80 |
| Elk-1 | Yes (HTTP) | 20.213.240.243 | 5601 |

1. **Elk Configuration**

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because:

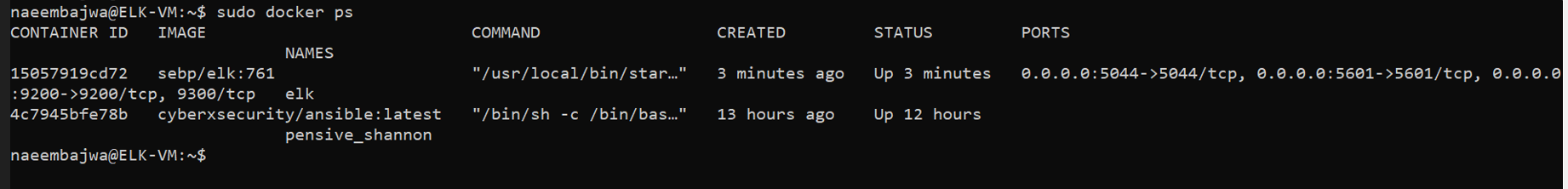
* Build and deployment is performed automatically, consistently and quickly
* Consistent, rapid configuration and deployment of virtual machines ensure all prescribed security measures can be scripted to minimise attack surfaces while enabling horizontal and elastic scaling by deployment to more or fewer virtual machines in a cluster as required to meet capacity demand.
* Facilitates OS and software updates

**Playbooks**

The three playbooks above implement the following tasks:

* Installs Docker, Python and Docker's Python Module
* Downloads and launches the DVWA Docker container
* Enables the Docker service
* Increase virtual memory to support the ELK stack
* Download and launch the Docker ELK container
* Downloads and installs Filebeat
* Enables and configures the system module and launches Filebeat.

The following screenshot displays the result of running docker ps after successfully configuring the ELK instance.



### **Target Machines and Beats Being Monitored**

This ELK server is configured to monitor the following machines:

* Web-1: 10.0.0.5
* Web-2: 10.0.0.6
* Web-3: 10.0.0.8

**Beats in Use**

We have installed the following Beats on these machines:

* Filebeat
* Metricbeat

These Beats allow us to collect the following information from each machine:

* Filebeat collects and ships (sends to ELK for collation, persistence, and reporting) logs from VMs running the Filebeat agent
* Metricbeat collects and ships system metrics from the operating system and services of VMs running the Metricbeat

### **Using the Playbooks**

In order to use the playbook, you will need to have an Ansible control node already configured. Assuming you have such a control node provisioned:

SSH into the control node and follow the steps below:

* Copy the playbook files to Ansible Docker Container.
* Update the Ansible hosts file “/etc/ansible/hosts” to include the following:

**[webservers]**

10.0.0.5 ansible\_python\_interpreter=/usr/bin/python3

10.0.0.6 ansible\_python\_interpreter=/usr/bin/python3

10.0.0.8 ansible\_python\_interpreter=/usr/bin/python3

**[elkservers]**

10.1.0.4 ansible\_python\_interpreter=/usr/bin/python3

Update the Ansible configuration file /etc/ansible/ansible.cfg and set the remote\_user parameter to the admin user of the web servers

* Run the playbook and navigate to Filebeat installation page on the ELK server GUI [http://[your.ELK-VM.External.IP]:5601/app/kibana] to check that the installation worked as expected.

1. **How to Use the Ansible Build [Running the Playbooks]**
2. Start an ssh session with the Jump Box ~$ ssh sysadmin@<Jump Box Public IP>
3. Start the Ansible Docker container ~$ sudo docker start <Ansible Container>
4. Attach a shell to the Ansible Docker container with the command ~$ sudo docker attach <Ansible Container Name>
5. Run the playbooks with the following commands:
   * ansible-playbook /etc/ansible/pentest.yml
   * ansible-playbook /etc/ansible/install-elk.yml
   * ansible-playbook /etc/ansible/roles/filebeat-playbook.yml

* Note that the Playbook 2 - install\_elk.yml configures only the server(s) listed as [elkservers] in /etc/ansible/hosts
* Similarly Playbook 3 - filebeat-playbook.yml configures the servers listed as [webservers] in /etc/ansible/hosts
* After running the playbooks and observing no errors in the output, navigate to Kibana to check that the installation worked as expected by viewing Filebeat and Metricbeat data and reports in the Kibana Dashboard
* Kibana can be accessed at [http://<elk-server-ip>:5601/app/kibana](https://github.com/luke-ozicyber/bootcamp/blob/main)

**Misc Questions:**

**Q.1 - Which file is the playbook? Where do you copy it?**

Ans: Playbooks have extension .yml and it is copied from the ELK container to the Virtual Machines.

**Q.2 - Which file do you update to make Ansible run the playbook on a specific machine? How do I specify which machine to install the ELK server on versus which to install Filebeat on?**

Ans: We update the “hosts” file located at /etc/hosts and add the IP address of the Virtual Machines under title “Webservers” so that the playbook runs the tasks on those machines.

In order to specify on which machine to install ELK server and on which to install Filebeat, we just change the value of the field called hosts in the playbook file, such as

hosts: elkservers

**Q.3 - Which URL do you navigate to in order to check that the ELK server is running?**

Ans: To check that the ELK server is running, you can can access your server by navigating to below URL

http://[your.ELK-VM.External.IP]:5601/app/kibana.

Use the public IP address of your new VM.