**Automated ELK Stack Deployment**

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

Graphical user interface, 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 Ansible Playbook file may be used to install only certain pieces of it, such as Filebeat.

Text

Description automatically generated

This document contains the following details:

- Description of the Topology

- Access Policies

- ELK Configuration

- Beats in Use

- Machines Being Monitored

- How to Use the Ansible Build

**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 the system from DDoS attacks by shifting attack traffic. The advantage of a jumpbox is to give access to the user from a single node that can be secured and monitored.

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

* Filebeat is a lightweight transport for forwarding and centralizing log data. Installed as an agent on your servers, Filebeat monitors the log files or locations that you specify, collects log events, and forwards them either to Elasticsearch or Logstash for indexing.
* Metricbeat takes the metrics and statistics that it collects and transports them to the output that you specify, such as Elasticsearch or Logstash. Metricbeat helps you monitor your servers by collecting metrics from the system and services running on a server.

The configuration details of each machine may be found below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Function** | **IP Address** | **Operating System** |
| Jumpbox | Gateway | 10.0.0.6 | Linux |
| ELK Stack | ElasticSearch Stack | 10.4.0.4 | Linux |
| Web-1 | Web Server | 10.0.0.4 | Linux |
| Web-2 | Web Server | 10.0.0.7 | Linux |

**### Access Policies**

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

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

* 74.133.14.133

Machines within the network can only be accessed by private IP addresses.

* The only machine that is allowed to connect to the ELK VM is through the Jump Box. The IP address to connect from the Jumpbox to the ELK VM is through 10.4.0.4, which is a private IP address.

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

|  |  |  |
| --- | --- | --- |
| **Name** | **Publicly Accessible** | **Allowed IP Addresses** |
| Jumpbox | SSH-22 – Yes, with my IP | 74.133.14.133 |
| ELK-Stack | No | 52.255.201.85 |
| Web-1 | No | Web LB 13.91.218.12 |
| Web-2 | No | Web LB 13.91.218.12 |
| Web LB | HTTP-80 - Yes | \* |
| ELK | Kibana-5601 Yes, with my IP | \* |
| ELK | HTTP API-9200 Yes, with my IP | 10.0.0.0/16 |

**### Elk Configuration**

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

* The main advantage in using Ansible as an automatic system is that it allows programs to run on a continuous basis without interruption or interaction from employees. It saves time and money by not having to continually be writing scripts.

The playbook implements the following tasks:

**Playbook 1: pentest.yml**

Pentest.yml is used to set up DMWA servers running in a Docker container on each of the web servers hown in the diagram above. It implements the following tasks:

* Installs Docker
* Installs Python
* Installs Docker’s Python Module
* Downloads and launches the DVWA Docker container
* Enables the Docker service

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

Install-elk.yml is used to set up and launch the ELK repository server in a Docker Container on the ELK server. It implements the following tasks:

* Installs Docker
* Installs Python
* Installs Docker’s Python Module
* Increase virtual memory to support the ELK stack
* Increase memory to support the ELK Stack
* Download and launch the Docker ELK container

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

Filebeat-playbook.yml is used to deploy Filebeat on each of the web servers so they can be monitored centrally using ELK services running on Elk-1. It implements the following tasks:

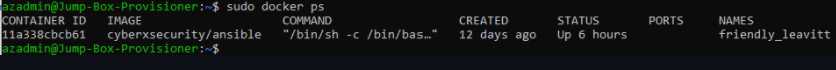
* Downloads and installs Filebeat
* Enables and configures the system module
* Configures and launches Filebeat

- ...

Text

Description automatically generated

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



**### Target Machines & Beats**

This ELK server is configured to monitor the following machines:

* Web-1 10.0.0.4
* Web-2 10.0.0.7
* Jumpbox Provisioner 10.0.0.6
* ELK 10.4.0.4

We have installed the following Beats on these machines:

* Filebeat
* Metricbeat
* ELK

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

* Filebeat collects data on filtered protocols that are set by the user. An example of such data collection can be exemplified by being able to manipulate the time duration of collecting the data, what type of data can be captured, such as 404 and 505 errors, and being able to filter out data derived from specific geographical locations. Also filebeat captures such events as logins to see who iss actively logging into the system. An example of a Filebeat is as follows:

Graphical user interface

Description automatically generated with low confidence

* Metricbeat is a lightweight shipper that you can install on your servers to periodically collect metrics/data from the operating system and from services running on the server. Metricbeat shows the metrics and statistics that is collects. An example of the data collected during a specific day is as follows:

Graphical user interface, application

Description automatically generated

**### Using the Playbook**

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 elk\_install.yml file to /etc/Ansible/roles/elk\_install.yml.
* Update the hosts file to include attribute, such as ELK, then include your destination IP of the ELK server directly below.
* Here is an example of a host file:

Text

Description automatically generated

**- Run the playbook, and navigate to** http://<[your\_elk\_server\_ip]>:5601/app/kibana to check that the installation worked as expected. An example of the start screen is as follows:

Graphical user interface, application

Description automatically generated

* Copy the playbook files to Ansible Docker Container.
* Update the Ansible hosts file /etc/ansible/hosts to include the following:
* 
* Update the Ansible configuration file /etc/ansible/ansible.cfg and set the remote\_user parameter to the admin user of the web servers.

**Running the Playbooks:**

1. Start an ssh session with the Jumpbox - **~$ ssh <username>@<Jump Box Public IP>**
2. Start the Ansible Docker container - **~$ sudo docker start <Ansible Container>**
3. Attach a shell to the Ansible Docker container with the command **~$ sudo docker attach <Ansible Container Name>**
4. Run the playbooks with the following commands:

* **Ansible-playbood /etc/ansible/pentest.yml**
* **Ansible-playbook /etc/ansible/install-elk.yml**
* **Ansible-playbook /etc/ansible/roles/filebeat-playbook.yml**

1. Note that the Playbook 2 – **install\_elk.yml** configures only the servers listed as **[ELK]** in **/etc/ansible/hosts**
2. Similarly Playbook 3 – **filebeat-playbook.yml** configures the servers listed as **[webservers]** in **/etc/ansible/hosts**
3. 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 data and reports in the Kibana Dashboard.

Graphical user interface, application, Word

Description automatically generated

1. Kibana can be accessed at http://<elk-server-ip>:5601/app/kibana