**Automated ELK Stack Deployment**

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

ELK-Stack Diagaram.drawio

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 YAML file may be used to install only certain pieces of it, such as Filebeat.

- filebeat-config.yml

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 efficient, in addition to restricting unwanted access to the network.

- Load balancers protect against attacks like DDos by maintain an even distribution.

- The jump box provides a one source access point that you can monitor any traffic in and out.

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

-Filebeat monitors log files or locations that you specify and puts the data in a centralized location.

-Metricbeat takes the statistics collected and sends them to your specified management system.

The configuration details of each machine may be found below.

| Name | Function | IP Address | Operating System |

|:--------:|:----------:|:----------:|:----------------:|

| Jump Box | Gateway | 10.0.0.4 | Linux |

| Web1 | Web Server | 10.0.0.5 | Linux |

| Web2 | Web Server | 10.0.0.6 | Linux |

| ELK | Monitor | 10.1.0.4 | Linux |

**Access Policies**

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

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

- 104.211.30.203

Machines within the network can only be accessed by going through SSH.

- The Elk machine can be accessed through the ansible container.

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

| Name | Publicly Accessible | Allowed IP Addresses |

|----------|---------------------|----------------------|

| Jump Box | Yes | 10.0.0.4 |

| Web1 | No | 10.0.0.5 |

| Web2 | No | 10.0.0.6 |

| ELK | No | 10.1.0.4 |

**Elk Configuration**

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

- its more efficient and minimizes risk for error.

The playbook implements the following tasks:

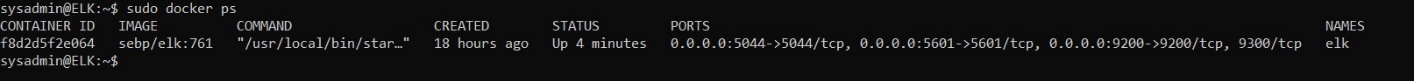
- Install Docker.io

- Install Python3

- Extend the memory

- Download and launch container

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:

- 10.0.0.5 and 10.0.0.6

We have installed the following Beats on these machines:

- Filebeat and Metricbeat

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

- Filebeat collects logs and detects changes in data

- Metricbeat analyzes metric stats from CPU, memory, etc.

**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 install-elk.yml file to ansible.

- Update the hosts.yml file to include the correct IP address

- Run the playbook, and navigate to http://<External.IP>:5601/app/kibana to check that the installation worked as expected.