## Automated ELK Stack Deployment

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

Images/network.diagram.png

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

Scripts/filebeat-playbook.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 available, in addition to restricting access to the network.

In terms of security, load balancers ensure maximum availability of service through distributing the network traffic between multiple servers to ensure that no one server is under too much demand by spreading the workload evenly. In addition to increasing application efficiency and responsiveness, load balancing also implements a layer of redundancy in the event of which one or more servers may be unavailable, network traffic will be rerouted to place a higher workload/stress on the available servers to ensure maximum application availability.

The deployment of a Jump Box in this network is to channel all outside traffic through a monitored VM with strict access controls in place to eliminate exposure of our elk and web vm’s to the internet in an effort to mitigate outsider threats from exploiting our private network. This is accomplished by configuring our web vm’s to only receive traffic from our Jump Box

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

Filebeat collects log data and events from our web server which it then forwards to logstash for further inspection

Metricbeat collects metrics and statistics from services running on our web server and forwards the data to logstash for further examination

The configuration details of each machine may be found below.

| Name | Function | IP Address | Operating System |

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

| Jump-Box | Gateway | 10.2.0.8 | Linux |

| Web-1 | Hosts Web Server | 10.2.0.5 | Linux |

| Web-2 | Hosts Web Server | 10.2.0.6 | Linux |

| ELK-VM | Hosts ELK Stack | 10.3.0.4 | 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 addresses:

-Local machine public IP

Machines within the network can only be accessed by SSH.

Only the Jump Box machine and local machine can access the ELK-VM

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

| Name | Publicly Accessible | Allowed IP Addresses |

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

| Jump-Box | Yes | Local Machine IP |

| Web-1 | No | 10.2.0.8 |

| Web-2 | No | 10.2.0.8 |

| ELK-VM | Yes | 10.2.0.8, Local Machine IP |

### Elk Configuration

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

Ansible allows for quick deployment of previously configured VM’s

The playbook implements the following tasks:

-install docker.io- installs the docker package

-install python3.pip- verifies the use of python3

-install docker module- allows execution of docker module commands

-increase virtual memory- allows ELK to maximize virtual memory use

-download and launch docker/enable service- installs and boots service automatically

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

Images/docker.ps

### Target Machines & Beats

This ELK server is configured to monitor the following machines:

-Web-1, Web-2

We have installed the following Beats on these machines:

- Web-1, Web-2

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

Filebeat collects system logs (system) and parses logs created by the system logging service. Metricbeat collects docker metrics from the system services running on the server

### 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 config file to ansible container.

- Update the config file to include hosts IP of ELK machine

- Run the playbook, and navigate to filebeat installation page to check that the installation worked as expected.

\_TODO: Answer the following questions to fill in the blanks:\_

- \_Which file is the playbook? Where do you copy it?\_

- \_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?\_

Filebeat-playbook.yml is the playbook. TO specifiy which machine to install the ELK server on, the filebeat-config file must be edited and the hosts IP must be changed to the IP of the ELK-VM. To specify which machines to install filebeat on, the playbook must specify the host group as “webservers” with the IP addresses of the host machines being added to the “webservers” group in the hosts file of the ansible container.

- \_Which URL do you navigate to in order to check that the ELK server is running?

<http://(ELK-VM> IP:5601)/app/kibana

\_As a \*\*Bonus\*\*, provide the specific commands the user will need to run to download the playbook, update the files, etc.\_

Ansible-playbook filebeat-playbook.yml